



# Using Agent Based Distillations to Support Land Warfare Studies

*Dion Grieger*

**Land Operations Division  
Electronics and Surveillance Research Laboratory**

DSTO-GD-XXXX

## **ABSTRACT**

Agent Based Distillations (ABDs) are low resolution abstract models that seek to address three key areas that traditional land combat models tend to neglect - non-linear behaviour, co-evolving landscapes and intangibles. This report examines the current suite of ABDs and their effectiveness as a tool to support existing simulations. Results from studies on Manoeuvre in the Littoral Environment, Uncertainty in the Battlefield, and Reconnaissance and Surveillance are presented, together with some possible improvements to the movement algorithms of existing ABDs

## **RELEASE LIMITATION**

*Approved for public release*



*Published by*

*DSTO Electronics and Surveillance Research Laboratory  
PO Box 1500  
Edinburgh South Australia 5111 Australia*

*Telephone: (08) 8259 5555  
Fax: (08) 8259 6567*

*© Commonwealth of Australia 2002  
AR- XXX-XXX  
December 2002*

**APPROVED FOR PUBLIC RELEASE**



# Using Agent Based Distillations to Support Land Warfare Studies

## Executive Summary

Agent Based Distillations (ABDs) are low resolution models that can be used to explore a large parameter space in a short period of time. They are intended to focus on areas that are thought to be deficient in existing models: non-linear behaviour (where a small change in one parameter can lead to a large change in another), co-evolving landscapes (which describes the changing nature of the battlefield) and intangibles like morale, training and bravery. They are not intended to produce conclusive evidence but rather to guide further analysis with other models and methods.

The first part of this report examines the features of the current suite of ABDs and their similarities, differences and limitations. The second part of the report highlights some problems with the movement algorithms in current ABDs and suggests a possible alternative. This section is followed by the results from a number of case studies that have been conducted to examine the utility of using ABDs as a tool to support existing methods that study land warfare. The first case study aims to answer questions relating to force mix in the littoral environment. The second attempts to replicate an existing study on battlefield uncertainty using ABDs as a model for land warfare. The third study examines the impact of reconnaissance and surveillance on battlefield survivability while the final case study details an attempt to examine a new concept involving entry by air and sea.

There appears to be no one ABD that is superior to any other, as each provides different features that may be useful depending on the scenario at hand. The results of the case studies seem to suggest that the level of detail of the current suite of ABDs may be too low to allow their results to be compared with other models for many of the current concepts being explored by LOD. However, with the continued development of these models there does appear to be a place for ABDs as a tool to support our existing methods.



## Authors

### **Dion Grieger**

Land Operations Division

*Dion Grieger is an Industry Based Learning Student at LOD, DSTO. He is currently studying at the University of Adelaide and will be completing a double major in Applied Mathematics and Computer Science in 2002.*

---



<b>DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION</b> <b>DOCUMENT CONTROL DATA</b>					
				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)	
2. TITLE  Using Agent Based Distillations to Support Land Warfare Studies			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)  Document (U) Title (U) Abstract (U)		
4. AUTHOR(S)  Dion Grieger			5. CORPORATE AUTHOR  Electronics and Surveillance Research Laboratory PO Box 1500 Edinburgh South Australia 5111 Australia		
6a. DSTO NUMBER DSTO-GD-XXXX		6b. AR NUMBER AR-XXX-XXX		6c. TYPE OF REPORT General Document	
7. DOCUMENT DATE					
8. FILE NUMBER D9505-23-26	9. TASK NUMBER DST 00/051	10. TASK SPONSOR Chief, LOD		11. NO. OF PAGES 79	12. NO. OF REFERENCES 27
13. URL on the World Wide  <a href="http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-xxxx.pdf">http://www.dsto.defence.gov.au/corporate/reports/DSTO-GD-xxxx.pdf</a>				14. RELEASE AUTHORITY  Chief, Land Operations Division	
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT  <p style="text-align: center;"><i>Approved for public release</i></p>					
OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111					
16. DELIBERATE ANNOUNCEMENT  No Limitations					
17. CITATION IN OTHER DOCUMENTS Yes					
18. DEFTEST DESCRIPTORS  Modelling Army planning Military operations Land combat					
19. ABSTRACT Agent Based Distillations (ABDs) are low resolution abstract models that seek to address three key areas that traditional land combat models tend to neglect - non-linear behaviour, co-evolving landscapes and intangibles. This report examines the current suite of ABDs and their effectiveness as a tool to support existing simulations. Results from studies on Manoeuvre in the Littoral Environment, Uncertainty in the Battlefield, and Reconnaissance and Surveillance are presented, together with some possible improvements to the movement algorithms of existing ABDs					

# Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>1.1 Project Albert.....</b>	<b>1</b>
<b>1.2 Agent Based Distillations.....</b>	<b>1</b>
<b>1.3 Outline of this Report .....</b>	<b>2</b>
<b>2. CURRENT SUITE OF AGENT BASED DISTILLATIONS.....</b>	<b>3</b>
<b>2.1 ISAAC .....</b>	<b>3</b>
2.1.1 ISAACA Parameters .....	3
2.1.2 Command Structure in ISAAC .....	5
2.1.3 Data Collection in ISAAC.....	6
<b>2.2 EINStein .....</b>	<b>8</b>
2.2.1 Terrain in EINStein .....	10
2.2.2 Data Collection in EINStein .....	11
2.2.3 Additional Features in EINStein .....	13
<b>2.3 MANA.....</b>	<b>14</b>
2.3.1 Terrain in MANA .....	17
2.3.2 Data Collection in MANA .....	17
2.3.3 Future Enhancements for MANA .....	18
<b>2.4 Socrates .....</b>	<b>18</b>
<b>2.5 Archimedes and Pythagoras .....</b>	<b>20</b>
<b>2.6 Bactowars.....</b>	<b>21</b>
<b>2.7 Comparison of the Current Suite of Agent Based Distillations .....</b>	<b>22</b>
<b>3. MOVEMENT WITHIN ABDS.....</b>	<b>23</b>
<b>3.1 EINStein and MANA Movement Algorithms .....</b>	<b>24</b>
3.1.1 One-Dimensional Scenario Analysis .....	24
3.1.2 Discussion of Behaviour .....	26
<b>3.2 Alternative Movement Algorithm.....</b>	<b>27</b>
3.2.1 Threshold Analysis.....	27
3.2.2 Inverse Distance Estimators .....	28
3.2.3 Relative Distance Estimators .....	30
3.2.4 Utility Curves.....	31
3.2.5 Cumulative Penalties .....	32
3.2.6 One Dimensional Scenario Analysis.....	33
3.2.7 Two-Dimensional Scenario Analysis.....	34
<b>3.3 Extensions.....</b>	<b>36</b>
3.3.1 Stochastic Movement .....	36
3.3.2 The Flag.....	37
<b>3.4 Summary and Future Research .....</b>	<b>39</b>
<b>4. CASE STUDIES .....</b>	<b>40</b>
<b>4.1 Manoeuvre Operations in the Littoral Environment .....</b>	<b>40</b>
4.1.1 The Scenario .....	40
4.1.1.1 Parameter Specification .....	41
4.1.1.2 HIMARS Modelling .....	42
4.1.2 Results .....	42

4.1.2.1	Interactive Playback Mode.....	42
4.1.2.2	One-way Sensitivity Analysis.....	43
4.1.2.3	Fitness Landscape Analysis .....	44
4.1.2.4	Dispersion versus Speed.....	47
4.1.3	Summary and Conclusions .....	48
<b>4.2</b>	<b>Battlefield Uncertainty.....</b>	<b>49</b>
4.2.1	Introduction.....	49
4.2.1.1	Background .....	49
4.2.1.2	Motivation .....	49
4.2.2	Experiment .....	50
4.2.2.1	Surrogates for Uncertainty .....	50
4.2.2.2	Experiment Design.....	51
4.2.3	Results .....	52
4.2.4	Conclusions .....	54
<b>4.3</b>	<b>Reconnaissance and Surveillance.....</b>	<b>55</b>
4.3.1	Introduction.....	55
4.3.2	Workshop Results.....	57
4.3.3	Additional Data Farming .....	60
4.3.4	Other Modelling .....	64
4.3.5	Conclusions .....	65
<b>4.4</b>	<b>Support for Headline Experiment 2001 .....</b>	<b>66</b>
4.4.1	MANA Results.....	66
4.4.2	Socrates Results.....	69
4.4.3	Conclusions .....	71
<b>5.</b>	<b>SUMMARY</b>	<b>72</b>
<b>6.</b>	<b>ACKNOWLEDGEMENTS.....</b>	<b>73</b>
<b>7.</b>	<b>REFERENCES.....</b>	<b>74</b>

## List of Figures

Figure 1: ISAAC screen capture .....	5
Figure 2: Results being displayed using the Albert VisTool. The vertical axis is chosen as some function of the measures of effectiveness. ....	7
Figure 3: Editing agent parameters in EINStein .....	9
Figure 4: EINStein communications matrix .....	10
Figure 5: EINStein Interactive Run Mode.....	11
Figure 6: Results displayed after EINStein Multiple Time Series Mode .....	12
Figure 7: Results displayed after EINStein 2-Parameter Fitness Landscape Mode.....	12
Figure 8: EINStein Inter-Squad matrix.....	13
Figure 9: EINStein grenade parameters .....	14
Figure 10: Editing general squad properties in MANA.....	15
Figure 11: Editing personalities in MANA .....	16
Figure 12: Editing squad ranges in MANA .....	17
Figure 13: Socrates Viewer.....	19
Figure 14: Bactowars scenario screen .....	21
Figure 15: Simplified One-Dimensional Scenario.....	24
Figure 16: Utility Curves Generated by the New Penalty Function .....	32
Figure 17: Variation in Paths Generated by Alternative Movement Algorithms .....	35
Figure 18: Effect of the Second Flag Alternative.....	38
Figure 19: Effect of the Third Flag Alternative.....	39
Figure 20: Effect of the Fourth Flag Alternative.....	39
Figure 21: Snapshots of baseline scenario simulation.....	44
Figure 22: Force size and dispersion level .....	45
Figure 23: Sensor range and probability of kill.....	46
Figure 24: Coordinated actions. Numbers represent the relative probability that two or three entities will respond to the same detection .....	51
Figure 25: Dispersed Scenario and Grouped Scenario.....	52
Figure 26: Baseline Scenario Elements .....	56
Figure 27: Number of Recon Assets vs. Combat Threshold.....	58
Figure 28: Number of Recon Assets vs. Weapon Range.....	59
Figure 29: Number of Recon Assets vs. Communications Weight (0-1) .....	60
Figure 30: Number of Recon Assets vs. Communications Weight (0-0.4).....	61
Figure 31: Effect of Trading Strike for Recon Assets in Different Terrain Types .....	62
Figure 32: Effect of Adding Strike Assets in Different Terrain Types .....	63
Figure 33: Effect of Adding More Indirect Fire.....	63
Figure 34: Visual Description of the Four MANA Scenarios .....	67
Figure 35: Effectiveness of Increasing Number of POE against Red COA .....	69
Figure 36: Effect of Firepower Superiority on Best COA.....	69
Figure 37: Single and Multiple Insertion Points in Socrates.....	70

## List of Tables

Table 1: Comparison of ABDs .....	22
Table 2: Penalty function variables .....	25
Table 3: Penalty function component calculations .....	25
Table 4: New Penalty Function Results on the Simplified One-Dimensional Scenario	33
Table 5: Effect of the r parameter on penalty components .....	33
Table 6: Relative distance movement algorithm paths .....	34
Table 7: Movement Selection Scheme Probability Distributions .....	37
Table 8: Major physical characteristics .....	41
Table 9: LER for different modes of Red movement .....	47
Table 10: Effect on LER as uncertainty increases .....	53
Table 11: Excursions from the Baseline .....	56
Table 12: Baseline Model Parameters -- ISAAC Attributes and Personalities .....	57
Table 13: Parameter Modifications for Terrain Type .....	61
Table 14: Comparison of Blue success across Excursions as indicated by the LER .....	65
Table 15: MANA Scenario Analysis .....	67
Table 16: Effectiveness of Blue COA against Red COA .....	68

## Glossary of Acronyms

ABD	agent based distillation
AEF	Army Experimental Framework
AO	area of operations
ARH	armed reconnaissance helicopter
CA	cellular automata
CATDC	Combined Arms Training and Development Centre
COA	course of action
DP	decisive point
DSTO	Defence Science and Technology Organisation
DTA	Defence Technology Agency
EAS	Entry by Air and Sea
EINStein	Enhanced ISAAC Neural Simulation Toolkit
HE	Headline Experiment
HIMARS	high mobility artillery rocket system
ISAAC	Irreducible Semi-Autonomous Adaptive Combat
ISAACA	ISAAC Agent
LAV	light armoured vehicle
LER	loss exchange ratio
MANA	Map Aware Non-uniform Automata
MCCDC	Marine Corps Combat Development Command
MHPCC	Maui High Performance Computing Center
MOE	measure of effectiveness
MOLE	manoeuvre operations in a littoral environment
POE	points of entry
R&S	reconnaissance and surveillance
RTA	Restructuring the Army
SA	situation awareness
SIRE	Simulation of Intelligent Reactive Entities
USMC	United States Marine Corps

# 1. Introduction

## 1.1 Project Albert

Project Albert is a United States Marine Corps (USMC) research effort that attempts to assess the general applicability of the concept of 'Operational Synthesis' [1] to land warfare. A central idea is that land warfare can be thought of as a complex adaptive system [2], which is a non-linear dynamical system of many interacting agents continuously adapting to a changing environment. Thus, Project Albert seeks to address three key areas: non-linear behaviour (where small changes create disproportionate responses); co-evolving landscapes (which characterise the changing battlefield) and intangibles (such as morale, discipline and training) for which conventional land combat analysis models are thought to be poor at investigating.

The New Zealand Defence Technology Agency (DTA) has been active recently in using the tools within Project Albert to assist in restructuring their combat force [3]. The Australian Army and the Australian Defence Science and Technology Organisation (DSTO) have subsequently become collaborators within the Project Albert research program.

Operational synthesis is the process of using all available tools and resources to synthesize information to answer questions pertaining to the three key areas mentioned above. For the example of land warfare these tools may include wargames, equations, field trials and agent based distillations (ABD's). Each tool should be used for what it is particularly useful for and to obtain results that other tools cannot. By combining all of these results we hope to gain a greater understanding of the problem at hand.

Data farming [4] describes the process of gaining the information required from each tool. It is essentially the collection of large amounts of data that can be sifted through in order to find the areas of interest. Often it may involve searching extremely large data sets and homing in on one particular area by completing more trials, runs or simulations.

Operational synthesis and data farming need not only be applied to ABD's as described in this report, but to all areas of research.

## 1.2 Agent Based Distillations

Agent based distillations are low-resolution abstract models, used to explore questions associated with land combat operations in a short period of time. Being deliberately low resolution means that the detailed physics of combat are largely ignored. Typically this involves assigning simple numerical values for characteristics such as speed, sensor, communication and weapon ranges, lethality and vulnerability. Being agent-based means that only simple behavioural rules need to be assigned. This is generally achieved by assigning 'personalities' to the agents by way of relative weightings to various elements on

the battlefield (friendly and enemy agents, notional 'flags', terrain features, and so on) and a linear penalty function to determine the agent's next move. Various 'meta-personalities' can also be assigned which moderate the agent's default personality if certain threshold constraints are exceeded from time to time.

Thus the scenario is much less scripted than that of traditional war-games, the idea being to allow a focusing of thought on the essential elements of the systems, which typically is the dynamic interaction of agents on the battlefield. Advances in computing power can then be exploited to produce a significant volume of data. This process, described earlier as data farming, allows extensive parameter excursions to be performed, both in terms of variations in physical characteristics and tactics (behavioural characteristics), from the baseline scenario. The fact that such large parameter spaces can be explored allows the user to investigate any non-linear behaviour and synergies in the system. The farmed data can also be used to perform statistical analyses to test the significance of the properties observed.

This is significantly different to traditional war-games where simulations may take weeks or months to set up and run. The trade-off is that the level of detail available to the user is sacrificed when ABD'S are used. This level of abstraction means that there is a risk that the results may be invalid. This is where the art of operational synthesis can be used to focus ideas for further analysis with more detailed models to test the hypotheses provided by ABD's.

The models referred to in this report are all examples of ABD's because they all have the characteristic of having agents that are controlled by decision-making algorithms. They also all fall into the category of cellular automation (CA) models. These types of models have been used before in other areas of science [5] and the well known von Neumann's "Game of Life" is an example of a CA model [6]. CA models have also been used before for military exploration [7] but in a much more limited fashion than that envisioned for Project Albert [2].

### **1.3 Outline of this Report**

This report is a consolidation of the work the author performed at DSTO during the course of 2001. Much of this work has been conducted with other members of DSTO, and some of this work has been documented in other formal reports or publications. Appropriate references are provided in the text below.

The first part of this report examines the features of the current suite of ABD's and their similarities, differences and limitations. The second part of the report highlights some problems with the movement algorithms in current ABD's and suggests a possible alternative. This is followed by the results from a number of case studies that have been conducted to examine the utility of using ABD's as a tool to support existing methods that study land warfare. The first of these case studies aims to answer questions relating to

force mix in the littoral environment. The second attempts to replicate an existing study on battlefield uncertainty using ABD's as a model for land warfare. The third study examines the impact of reconnaissance and surveillance on battlefield survivability while the final case study details an attempt to examine a new concept involving entry by air and sea. Finally a summary containing opinions held by the author on the use of ABD's and their relevance to land warfare studies.

## 2. Current Suite of Agent Based Distillations

There has been a growing number of ABD's developed since the forming of Project Albert. The following sections provide an overview of the general characteristics of each one followed by a summary table comparing them with each other.

### 2.1 ISAAC

The first ABD developed as a result of Project Albert was ISAAC (Irreducible Semi-Autonomous Adaptive Combat) [2]. In this section I will be referring to ISAAC Version 1.8.6. Setting up a scenario in ISAAC is long and tedious compared to the process with its successor EINSTEIN, and the New Zealand model, MANA (both described below). Input files need to be created either via a standard text editor or by following the DOS prompts. As a user I found that the easiest way to create a scenario was to edit one of the default ISAAC scenarios using a text editor.

The basic element of ISAAC is an ISAAC Agent (ISAACA). ISAAC allows for two sides (typically friends and enemies) with up to 400 agents per side, which may be separated into up to ten squads. Squads are initially located in a user-defined location on a 150 by 150 (maximum) grid. One flag or objective may also be positioned for each team.

#### 2.1.1 ISAACA Parameters

The user can define a number of parameters for each agent, which govern how they move, shoot, sense and communicate. A sample shot of an ISAAC screen is shown in Figure 1. The battlefield grid is in the centre of the screen while the respective parameters for each side are shown either side of the battlefield. The first set of parameters defines the physical characteristics of each agent. The sensor range (S-range) is the maximum range at which an agent can sense other agents in its vicinity. Fire range (F-range) is the maximum range to which an agent can fire upon an enemy agent. Movement range (M-range) is the maximum number of grid squares an agent can move in any single time step. For example, if movement range is two an agent may move up to two squares from its original position, giving a total of 25 potential movement locations.

Threshold range (T-range) defines the radius of the area that is considered important when an agent is making a move. For example, an agent may have a sensor range of 20 but the user may only want the agent to move based on the information that is close, so the threshold range may be set to five. Communications range (C-range) is the maximum distance over which an agent can communicate (share information) with other friendly agents. Probability of Hit (p-shot) is the probability that an agent will hit an enemy agent that is within firing range. Maximum targets (MAX TGT) is the maximum number of targets that an agent can engage in any one time step.

The next set of parameters defines the personality weights (P-weights) of the agents. These weights govern the movement of an agent. Weight towards alive red, injured red (AR, IR) define an agent's propensity to move towards (positive) or away (negative) from alive red agents and injured red agents respectively. Weight towards alive blue, injured blue (AB, IB) define an agent's propensity to move towards (positive) or away (negative) from alive blue agents and injured blue agents respectively. While weight towards red goal, blue goal (RG, BG) define an agent's propensity to move towards (positive) or away (negative) from the red flag and the blue flag respectively.

In addition to these basic movement weights an agent may be assigned additional constraints. These are called meta-personalities and are designed to allow an agent's personality to continually change depending on what it sees in its local environment. The Advance parameter (ADV) defines the threshold number of friendly agents that must be within an agent's threshold range in order for that agent to advance toward the enemy flag. If this constraint is in use and the requirements are not met at any one time step then the agent will negate the weight assigned to moving towards the enemy flag.

The Cluster parameter (CLS) uses the idea that once an agent is surrounded by a threshold number of friendly agents it will no longer attempt to move closer to them. This is achieved by negating the personality weightings to alive and injured friendlies whenever the threshold is exceeded.

The Combat parameter (CBT) applies when an agent senses that it has less than a threshold advantage of surrounding friendly forces than enemy forces it will move away from the enemy. Similarly to the cluster constraint this is achieved by negating the weight to alive and injured enemies when ever the threshold isn't met.

There are also a number of minimum distance constraints. Minimum distance to friendly or enemy agents (R\_M, B\_M) and minimum distance to own flag (R\_G or B\_G), negates the corresponding weights if an agent comes within a certain threshold distance of the agent in question. For example, if the minimum distance to friendly parameter was set to five and an agent was only three away from another friendly agent it would negate any positive weighting towards that friendly agent.

There are some additional parameters that are not shown on the screen during a run but can be viewed and edited via the appropriate input file. The first of these is the defence

measure, which is the number of times an agent must be shot before transitioning from Alive to Injured or from Injured to Dead. Reconstitution is an optional parameter allowing an agent to be reconstituted to its alive-state after a certain number of time steps after being hit by an enemy. Another parameter is fratricide, which is the probability that a friendly agent will be hit by a shot intended for, but which misses, an enemy. Finally communications weight is the importance an agent will place on communicated information (number between 0 and 1).

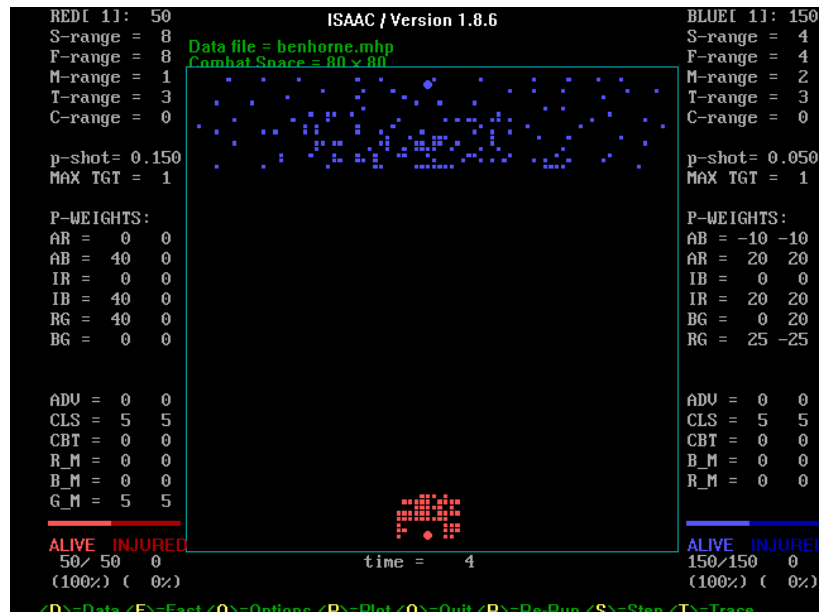


Figure 1: ISAAC screen capture

For each time step an agent may move to any of the squares within it's movement range or remain still. A penalty is calculated for each of the potential new locations based on the location of other agents and the agents personalities. The square with the lowest penalty is chosen as the new location. Movement within ISAAC is described in more detail in Section 3.

ISAAC has only very limited terrain features. The only form of terrain is an impassable object that may best be described as a building. Agents cannot see, shoot or travel through impassable terrain. Terrain blocks can be defined at the end of the input file by specifying each block's coordinates and the corresponding length and width.

### 2.1.2 Command Structure in ISAAC

There is a three level command structure in ISAAC. The user has the option of implementing these command structures to simulate a command and control hierarchy. The hierarchy consists of three different kinds of ISAACAs: elementary combatants which are standard ISAACAs with basic parameters as described above, local commanders that command and coordinate information flow among local groups of elementary combatants

and global commanders that have a global view of the battlefield and coordinate the actions of the local commanders.

Elementary ISAACAs are assigned to a local commander and are given two additional weights. One defines the propensity to stay close to their commander and the other defines the propensity to obey their commands.

The local commanders are given a surrounding command area that moves with them as they move throughout the battlefield. This command area is partitioned into smaller areas, which represent local goals that a commander can order his subordinates to move towards. A local commander identifies the partition that contains the smallest fractional difference between friendly and enemy forces and orders subordinate agents to move towards the centre of that partition. Movement of local commanders is based on the same personality weights listed earlier (they can be different to subordinate personalities) along with a propensity to help other local commanders and a propensity to obey orders issued by global commanders.

The global commander effectively has a complete view of the battlefield and issues orders to determine the direction in which each local commander should move. These orders are based on information obtained about the performance (attrition data and presence of enemy) of each local commander and his subordinates and the direction in which the global commander would like the local commanders to move towards.

### 2.1.3 Data Collection in ISAAC

As a user I would consider there to be four distinct data collection methods in ISAAC. The first of these is qualitative analysis, which involves simply watching a series of interactive runs on the screen while noting the emerging behaviours and characteristics. This method is often very effective if large parameter spaces do not need to be explored. It allows the user to see exactly what is happening and shows why, and how, a particular side is winning. Qualitative analysis is enhanced in ISAAC with the use of the Albert VisTool [8] where up to nine separate runs can be viewed at the same time via a tiled playback facility. The user also has the option of viewing a density playback of up to nine different runs. Runs are effectively played back on top of each other so the user can see which areas of the battlefield are generally occupied for a typical run.

The second method produces single run statistics. When prompted via the input file ISAAC will produce a statistical output file for a single run. This file contains attrition statistics (the number of remaining red and blue agents at each time step), goal statistics (the number of red and blue agents within certain ranges of both the red and blue flags), centre of mass statistics (keeps track of the centre of mass coordinates of both sides and of all agents) and cluster statistics (calculates the cluster size distribution at each time step).

This form of data collection is excellent when exact numbers are required. However its big downfall is that it only shows the results for one run when statistically we would prefer

many more runs [9]. It also makes it hard for the user to know why the results indicate certain trends without also performing some qualitative analysis as mentioned above.

Another method of data collection is via a web submittal to Maui High Performance Computing Centre (MHPCC) [10]. This method allows the user to submit an ISAAC input file to MHPCC and to explore up to five parameter spaces. The results can then be downloaded via the web and viewed using the Albert VisTool. This software package allows the user to generate 3 dimensional graphs for any combinations of the five input parameters selected. An example of a graph generated is shown in Figure 2.

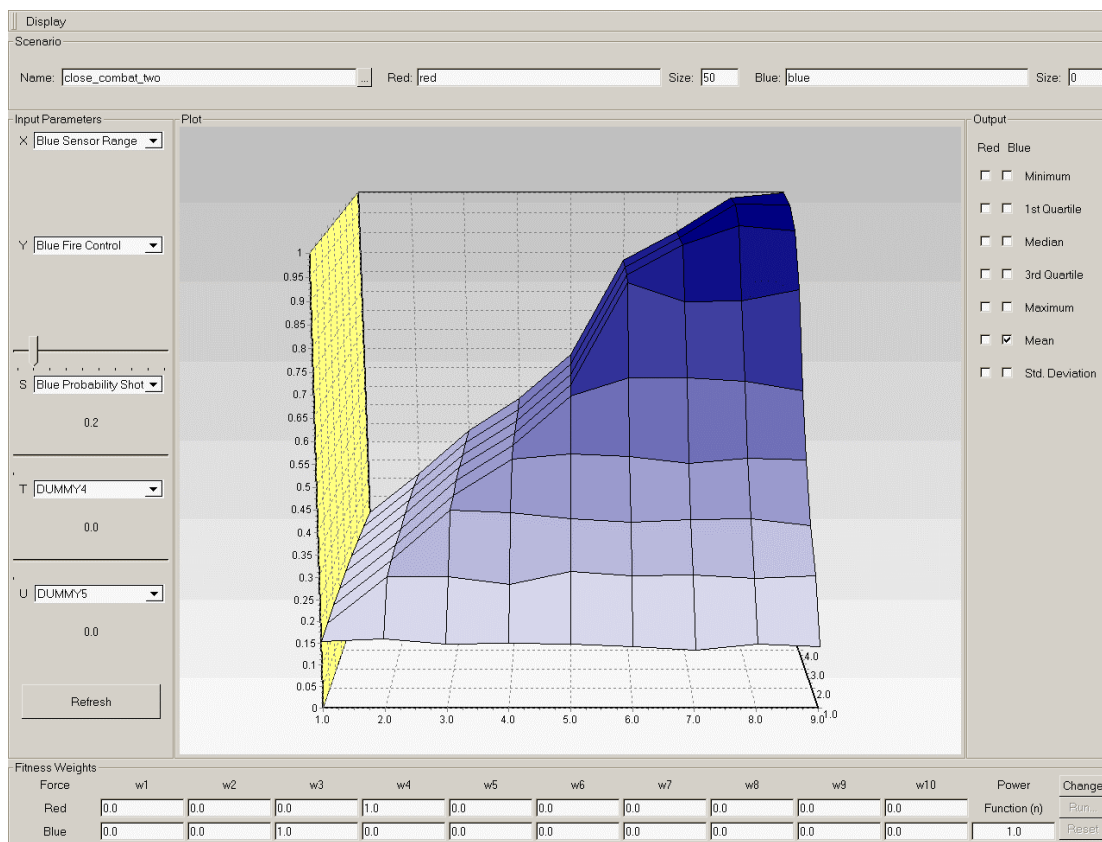


Figure 2: Results being displayed using the Albert VisTool. The vertical axis is chosen as some function of the measures of effectiveness.

The x and y-axes show the two parameters selected while the vertical axis is the measure of effectiveness. This measure can be chosen when the submittal is made and from a choice of the time to reach the flag, the number of casualties, the survival ratio, the distance to either flag or the number of fratricide hits to either side.

The benefit of this method of data collection is that such a large data set can be farmed and the results can be viewed quickly and easily. The downside is that there is often a wait to

get results back from MHPCC. The main reason for this wait is that all submittals are placed in a queue with other jobs (not just involving ABD's) so the time taken between submittal and execution can often be a couple of days. Compared to this waiting time execution time is negligible. Generally results can be expected back within two weeks although results for some runs have taken as little as 24 hours to return. When large amounts of data are produced it is often hard to determine the reason for certain trends and behaviours without going back and watching individual runs.

The final method of data collection in ISAAC is by using the genetic algorithm "Evolver". ISAAC has a separate program that allows the user to search for the best red force to conquer a fixed blue force. The user must define which red parameters can be varied and what the measure of effectiveness should be. ISAAC then "evolves" a red force that is best able to satisfy the objective within the constraints given.

## 2.2 EINSTEIN

EINSTEIN (Enhanced ISAAC Neural Simulation Toolkit) Beta Release Version 1.0.0.4 [11] is a user-friendly version of ISAAC which allows the user to set up and run a scenario in a Windows environment. It also has a number of additional features and parameters that will be highlighted in this section. The window in which squad properties are defined is shown in Figure 3.

Most of the parameters are the same as ISAAC but there are a few additions. Lethality contours give the option of varying the probability of hit of an agent depending on the distance to the target. Weighting towards an area is an additional personality weight that assigns an agent's propensity to move towards an area on the battlefield. This can be particularly useful for assigning an area of operation for a squad. This weighting can also be modified by the new meta-personality Minimum Distance to Terrain that works in the same way as the minimum distance constraints defined in Section 2.1.1.

**Edit::RED Agent Parameters**

**SQUAD**  
 Display Squad: 1 / 1  
 Squad Size: 200 / 200  
 SAVE Squad Data

**RANGES**

	Alive	Injured
Sensor Range	10	10
Fire Range	1	1
Threshold Range	10	10
Movement Range	0	0

**OFFENSE/DEFENSE**  
 Lethality Contours:  
☒ Fixed ☐ Normalized  
☐ User-Defined → P(R)  
 Prob(Hit): Alive 0.05, Injured 0.05  
 Max # Simul Tgts: 1  
 Defense Measure: 1

**PERSONALITY**  
 Randomize: ☐ Alive ☐ Injured  
 → Alive RED: 10, 10  
 → Alive BLUE: 40, 40  
 → Injured RED: 10, 10  
 → Injured BLUE: 40, 40  
 → RED Flag: 0, 0  
 → BLUE Flag: 0, 0  
 → LC: 0, 0  
 Obey LC: 0, 0  
 → Area: 0, 0  
 → Formation: 0, 0  
 → Terrain: 0, 0

**COMMUNICATIONS**  
☐ On ☒ Off  
 C[i][j]:  
 R: 0, W: 0, Alive: 0, Injured: 0

**FRATRICIDE**  
☐ On/Off R: 5 P(Hit): 0.5

**META-PERSONALITY**  
☐ On ☒ Off  
 Inter-Squad Weight Matrix S[i][j]  

	Use?	Alive	Injured
ADVANCE	<input checked="" type="checkbox"/>	3	3
CLUSTER	<input checked="" type="checkbox"/>	9	9
COMBAT	<input checked="" type="checkbox"/>	1	1
HOLD	<input type="checkbox"/>	0	0
PURSUIT - I	<input type="checkbox"/>	0	0
PURSUIT - II	<input type="checkbox"/>	0	0
RETREAT	<input type="checkbox"/>	0	0
SUPPORT-I	<input type="checkbox"/>	0	0
SUPPORT-II	<input type="checkbox"/>	0	0
Min D/RED	<input type="checkbox"/>	0	0
Min D/BLUE	<input type="checkbox"/>	0	0
Min D/RFlag	<input type="checkbox"/>	0	0
Min D/Terrain	<input type="checkbox"/>	0	0
Min D/Area	<input checked="" type="checkbox"/>	0	0

 RECONSTITUTION:  
☐ On/Off Recon-Time: 10

OK Cancel

Figure 3: Editing agent parameters in EINSTEIN

The weighting towards a formation parameter assigns an agents propensity to stay within a certain distance of the centre of mass of all friendly agents. Similarly the weighting towards terrain parameter assigns an agents propensity to move towards any terrain blocks on the battlefield. This is useful for forcing agents to stay in a well-protected area or forcing them to travel on (or off) a road. This weighting can also be modified by the new meta-personality Minimum Distance to Area that works in the same way as the minimum distance constraints defined in Section 2.1.1. The communications matrix ( $C[i][j]$ ) is shown in Figure 4 and allows the user to decide what squads can communicate with each other. If entry  $C_{i,j}$  is checked then squad  $i$  can receive information from squad  $j$ .

There are several new meta-personalities the first of which is the hold parameter. If an agent occupies a section of the battlefield that is locally occupied by a threshold number of friendly forces then it will set its movement to zero for that time step. The Pursuit I parameter means that if an agent senses that there are fewer than a threshold number of enemy agents nearby, it will ignore those agents. Similarly Pursuit II means that if an agent senses that there are fewer than a threshold number of enemy agents nearby, it will temporarily ignore all personalities except those towards enemy agents.

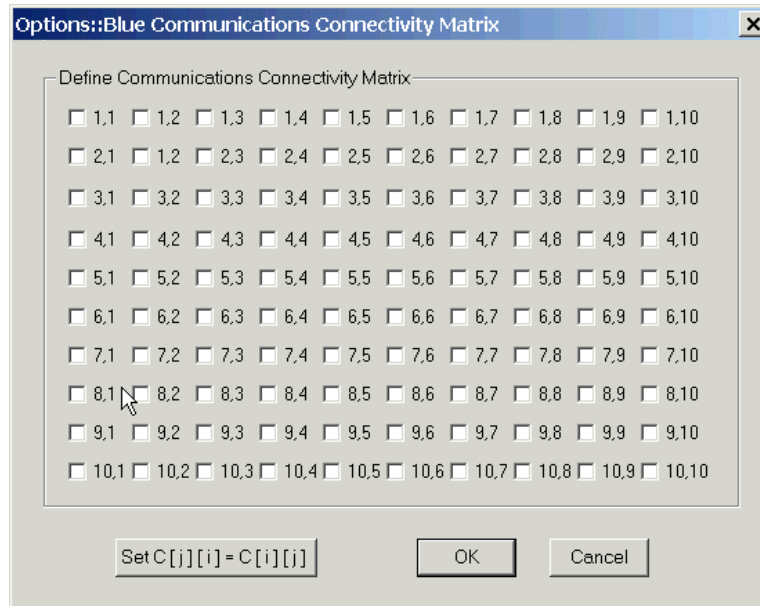


Figure 4: EINStein communications matrix

The retreat parameter allows an agent to retreat back to its own flag unless it is surrounded by a threshold number of friendly forces. Support I means that if an agent senses that there is more than a certain threshold number of nearby injured friendly agents it will temporarily ignore all other personality weights except those for injured friendly agents. While Support II allows an agent to temporarily ignore all other personality weights except those towards other alive and friendly agents to seek their support if it senses that there is more than a certain threshold of enemy agents.

### 2.2.1 Terrain in EINStein

The terrain features of EINStein have been improved significantly and allow the user to specify up to five types of terrain in addition to the default battlefield terrain. Type 1 can be thought of as a lava pit, agents can see across it but cannot pass through it. Type 2 is like a solid wall; agents can neither see through it nor pass through it. Types 3, 4 and 5 are user defined terrain types. The user can specify exactly what parameters will be affected when an agent is on the terrain block that allows the user to simulate just about any type of terrain. Parameters that can be modified include sensor, fire, threshold, movement and communications ranges, defence strength, probability of hit and visibility (probability of being detected).

With a combination of terrain types the user is able to construct a fairly detailed terrain structure containing, for example, roads, woods, jungles and rivers. It should be noted that currently there are some bugs with the terrain modifying parameters and some limitations, eg an agent cannot have an initial location on a piece of terrain. However the current structure does allow a simple scenario to be extended to explore the possibilities of agents being confronted by different types of terrain.

### 2.2.2 Data Collection in EINSTein

A significant benefit of EINSTein is its ability to collect and display large amounts of data in a relatively short period of time. There are four run modes available in EINSTein.

The Interactive Run Mode allows the user to view a single run at a time and has the option of pausing, stepping a single iteration at a time or restarting at any time. It is also possible to see the status of the agent (alive, injured or killed) through the use of a different coloured dot or icon. Figure 5 shows a screen capture of the battlefield half way through an EINSTein run. Injured blue agents are depicted as light blue dots while injured red agents appear as pink dots. Dead blue and red agents are represented by small black squares and black crosses, respectively.

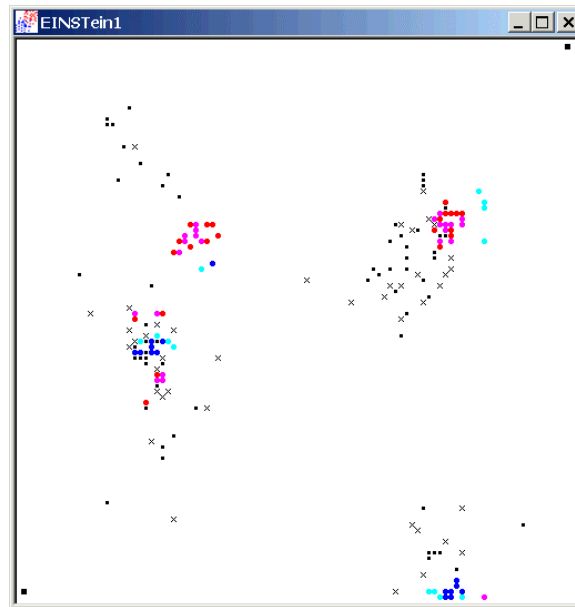


Figure 5: EINSTein Interactive Run Mode

The Multiple Time-Series Run Mode allows the user to produce a time series graph using data from a number of runs. As an example Figure 6 shows the number of remaining red and blue agents as a function of time. The error bars represent the average absolute deviation at each point. Other options for the measure of effectiveness ( $y$ -axis) include centre of mass distances to flag and the opposing force, cluster sizes, number of agents close to either flag, spatial entropy and territorial possession (how much of the battlefield each team “owns”).

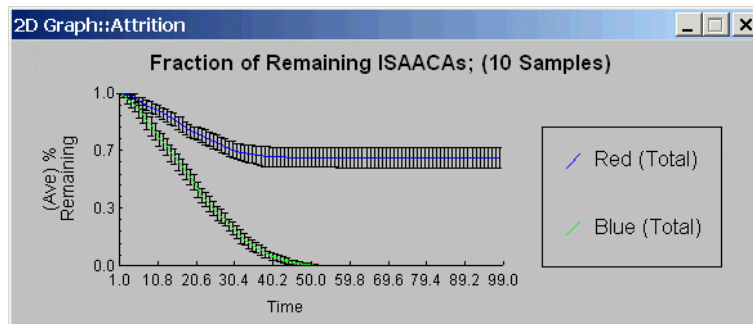


Figure 6: Results displayed after EINSTEIN Multiple Time Series Mode

2-Parameter Fitness Landscape Mode allows the user to “explore” a parameter space. 2 variables are chosen and an upper and lower bound and an increment size is defined for each. The user can then select how many runs to complete for each parameter combination and EINSTEIN will perform the required runs and save the output data in a text file. These results do not require a separate viewer and can be viewed within the EINSTEIN package. An example of a 2-D fitness Landscape graph is shown in Figure 7. This graph shows the red to blue survival ratio for varying sensor ranges and probability of hit for red. The available measures of effectiveness are the same as those for the ISAAC MHPCC web runs listed in Section 2.1.5.

One Sided Genetic Algorithm Run Mode is the same as the Genetic Algorithm mode in ISAAC but it is much easier to set up within the Windows environment.

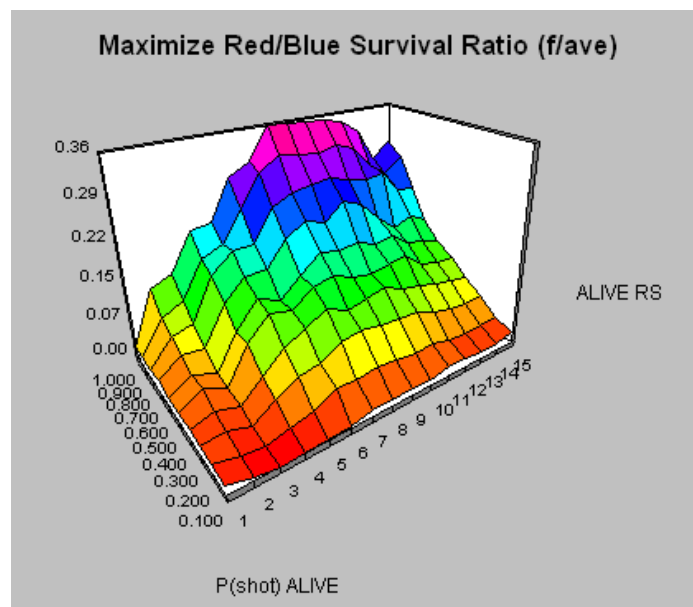


Figure 7: Results displayed after EINSTEIN 2-Parameter Fitness Landscape Mode

### 2.2.3 Additional Features in EINSTEIN

There are two other features worth mentioning that the author has found very useful when constructing scenarios. The first is the Inter-Squad Matrix, which is shown in Figure 8. This allows the user to modify the weights with which each squad reacts to other squads on the same side. This is particularly useful when a side is constructed of many squads that the user may want to work independently or in conjunction with other squads. This is especially important when heterogeneous squads are present, for example it may not be desirable for infantry type squads to be attracted to aerial squads and vice versa.

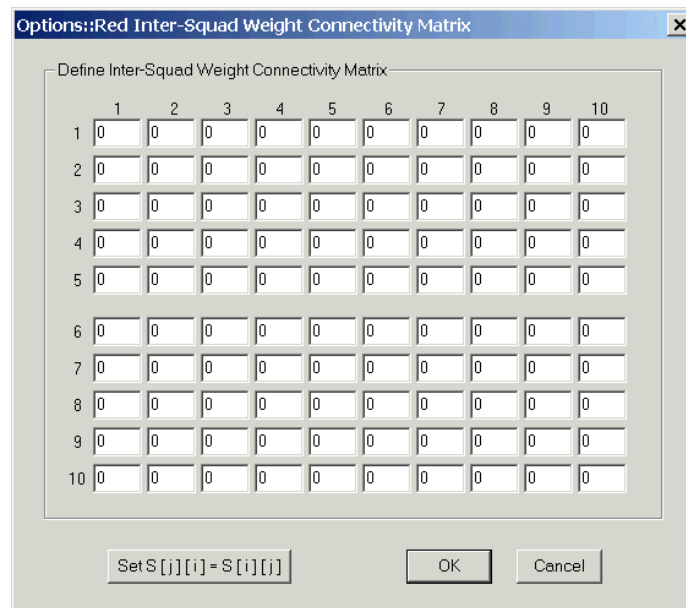


Figure 8: EINSTEIN Inter-Squad matrix

There have also been some good additions to the weapons parameters. EINSTEIN has introduced a grenade weapon in addition to the existing point-to-point weapons. The parameter box for a grenade is shown in Figure 9. The user can specify a minimum and maximum throwing range and can also target areas where the concentration of enemy forces is high. Like the point-to-point weapons, the accuracy of the grenade can be modified with distance.

**Edit::Blue Grenade Parameters**

**THROW PARAMETERS**

Minimum Throw Range  Maximum Throw Range

Prob that grenade will actually land at the (x,y) location it is thrown to ( $P(0) + P(1) + \dots + P(R_{max}) = 1$ )

	0	1	2	3	4	5	6	7	8	9	10
Probability	1	0	0	0	0	0	0	0	0	0	0

**PENALTY ASSESSMENT (i.e. decision criteria for where to throw grenade)**

Fratricide Tolerance (F) (will NOT target x-y if # friends > F)  Minimal Enemy Presence (E) (will NOT target x-y if # enemies < E)

**BLAST EFFECTS**

Blast Radius ( $R \leq 10$ )

**PROBABILITY OF HIT**

☒ P(Hit) = CONST  
 ☐ P(Hit) = P(Cookie Cutter Dist/f center)  
 ☐ P(Hit) = P(Euclidean Dist/f center)

(will use R=0 value)

	0	1	2	3	4	5	6	7	8	9	10
Probability(Hit)	0.4	0	0	0	0	0	0	0	0	0	0

OK Cancel

Figure 9: EINStein grenade parameters

## 2.3 MANA

The MANA model (Version 1.61) [12] was developed by New Zealand's Defence Technology Agency and was intended to address a broad range of problems. The MANA Users Manual [12] states that the motivation for designing MANA was a frustration with other models available to them.

MANA, like EINStein is very user friendly and runs in a Delphi Windows environment. As a result scenarios are quick and easy to set up. Movement in MANA is very similar to that in EINStein as it is based on personality weights towards friends and enemies although the movement algorithm is slightly different (see Section 3). Many of the parameters are exactly the same although there are some important additions to MANA that are very useful.

One of the key differences is the Situational Awareness (SA) map, which gives the user the option to give agents a view of the battlefield outside of their sensor range. It can be likened to communications in ISAAC and EINStein. Unfortunately the SA map is only used when there is no local information present that is there are no other agents within sensor range. The three windows screens used to set up a scenario in MANA are shown in Figures 10, 11 and 12. Any parameters that are different to those in ISAAC and EINStein or any new ones are commented on below.

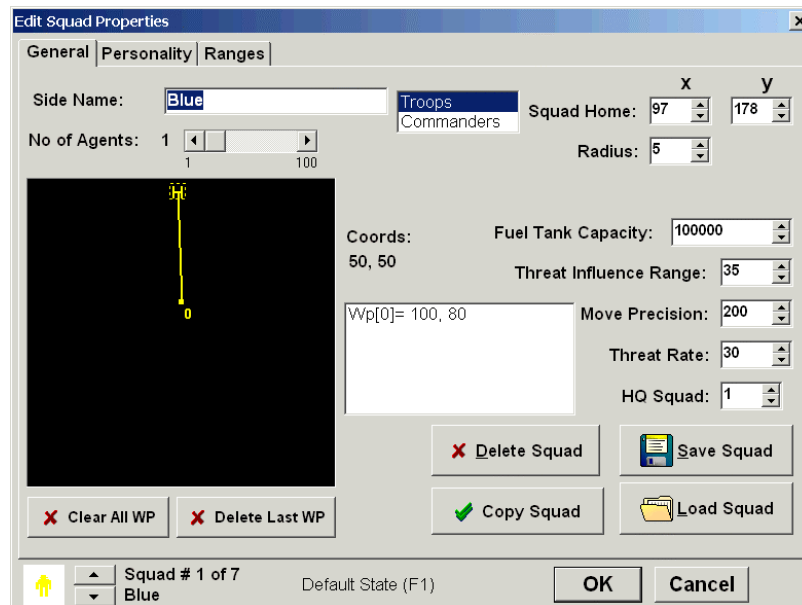


Figure 10: Editing general squad properties in MANA

The user can define multiple waypoints (WP), rather than just one flag as in ISAAC and EINSTEIN, to allow agents to follow a particular course. Troops are standard agents while commanders have the ability to order troops into areas where there is a lower density of opposing forces. This feature is still being developed and is not completely stable at the moment. It has similarities to local commanders in ISAAC and EINSTEIN.

Agents are given a fuel tank with an initial capacity. This does not strictly represent fuel, as an agent will not stop moving if the fuel tank is empty. It was intended that fuel could also represent something else like morale or stamina. The threat influence range is the distance to which agents can react to information on their SA map. Threat rate is the rate at which agents on the SA map fade. This can be likened to the memory of an agent. The headquarters squad (HQ Squad) is the squad that provides the information for an agent's SA map.

Along with the standard personality weights present in ISAAC and EINSTEIN MANA also has a weighting to distant friends. Distant friends are friendly agents outside an agent's sensor range but on their SA map. The easy terrain personality is the weighting towards yellow coloured terrain blocks (see Section 2.3.1). Enemy threat 1, 2 and 3 are weightings towards different enemy agents on the SA map. Agents of threat level 3 are the most recently spotted agents followed by those with threat level 2 and then 1. The minimum distance constraints are the same as in EINSTEIN except that the user can specify whether the minimum distance to friends is squad specific or if it applies to all friends.

**Edit Squad Properties**

General Personality **Ranges**

	Weighting	Min Distance		Constraints
w1 - Alive Friendlies	0	1	Alive/Inj Friends	0 Cluster
w3 - Injured Friendlies	0	Squad Only All Friends		2 Combat
w2 - Alive Enemies	0	0	Alive Enemies	0 Advance
w4 - Distant Friends	40			
w5 - Next Waypoint	30	35	Next Waypoint	
w6 - Enemy Final Goal	0	0	Enemy Goal	
w7 - Easy Terrain	0	0	Easy Terrain	
w8 - Enemy Threat 1	-20			
w9 - Enemy Threat 2	-20			
w10 - Enemy Threat 3	-20			

Zero Normalize Total: 130

State Options

Copy State Paste State

Duration [steps]: 0

Fallback to: Default State

Trigger States

- ☒ Default State
- ☒ Reach Waypoint
- ☐ Taken Shot
- ☐ Shot At
- ☐ Enemy Contact
- ☐ Squad Taken Shot
- ☐ Squad Shot At
- ☐ Squad En Contact
- ☐ Injured
- ☐ Squad Injured
- ☐ Squad Dead
- ☒ Attack
- ☐ Retreat
- ☐ Fuel Out
- ☐ State 15
- ☐ State 16

Squad # 1 of 7  
Blue

Default State (F1)

OK Cancel

Figure 11: Editing personalities in MANA

In the author's opinion the State Options and Trigger States (Figure 11) feature of MANA is the most beneficial and is what distinguishes MANA from other ABD's. Trigger states allow an agent to change its personality when a certain binary event occurs. The events are listed in the white box on the lower right side of Figure 11. For example, if an agent comes into contact with the enemy it may like to retreat or hide. This can be achieved by using the trigger state called enemy contact. This can be done individually or at a squad level where if any one agent sees an opponent then all agents who are friends of the other friendly agent will change state. The length of time an agent remains in this state is governed by the duration parameter. The user then has the option of controlling which state the agent should revert back to after the duration time expires.

Stealth is the likelihood of an agent being detected by the enemy. Agents of stealth 100 are invisible to enemy agents. Firepower is the same as probability of hit in ISAAC and EINSTEIN. The icon parameter defines which icon is used for each squad's representation on the screen. Allegiance defines which side a squad is on. Typically 0 is neutral, 1 is friendly and 2 is enemy. Neutral agents (not present in ISAAC and EINSTEIN) allow the user to simulate many scenarios where agents may not necessarily have any particular weighting to either side, eg civilians in peacekeeping scenarios. The threat parameter controls how enemy agents will react when the agent is seen on the enemy SA map. An agent's threat level decreases by one every threat rate time step. The fuel consumption rate is the amount of "fuel" used up per time step.

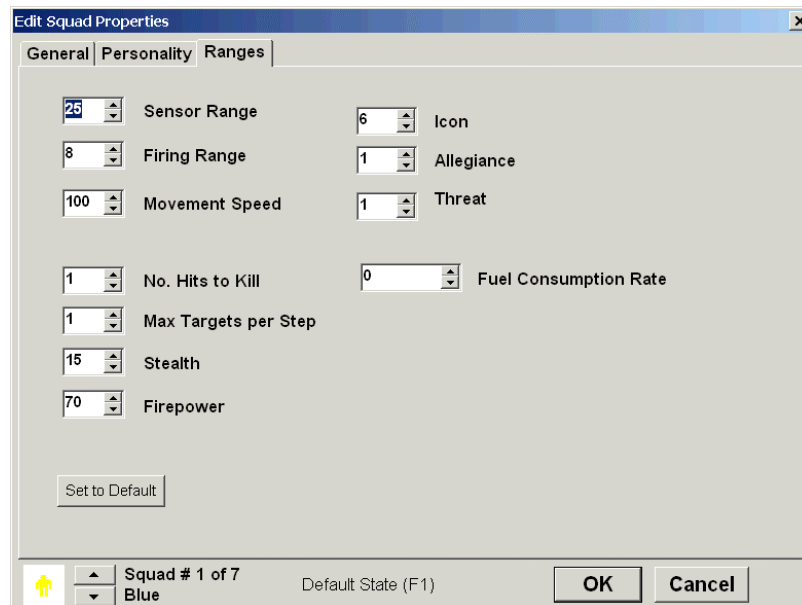


Figure 12: Editing squad ranges in MANA

### 2.3.1 Terrain in MANA

Terrain in MANA is currently limited to three types, passable (default), impassable (light grey) and roads (bright yellow). Terrain maps can be edited in any bitmap editor and then loaded into a scenario. MANA considers any colours other than the defined light grey and yellow as default terrain. Roads do not have an effect on any parameters of agents, eg they do not travel any faster on roads, but agents can be given a propensity to move towards or away from them (weighting to easy going terrain). Currently terrain is quite limited in MANA however future versions will incorporate more realistic terrain options (see Section 2.3.3).

### 2.3.2 Data Collection in MANA

One of MANA's biggest downfalls is the inability to explore large parameter spaces quickly and easily as can be done with ISAAC (via MHPCC) and EINSTEIN (via the 2 parameter fitness landscape mode). Currently MANA has two methods of data collection. Qualitative analysis can be done in a similar way to ISAAC and EINSTEIN. Data can be collected interactively by watching a series of single runs and noting the general trends. This is made much easier with MANA's user-friendly icons and the Windows environment.

MANA also has a Multiple Time Series Mode that allows the user to perform multiple runs of the same scenario. All results are then saved to a comma separated value file where they can be easily interpreted or graphed. Currently the only form of data captured is red losses and blue losses.

Whilst data collection in MANA is currently limited efforts are being made by the USMC and MHPCC together with DTA to create a web submittal system similar to ISAAC to allow large parameter spaces to be explored through an automated process at MHPCC.

### 2.3.3 Future Enhancements for MANA

MANA is continually being developed and an updated version is currently being coded. It is anticipated that the new version will include refuelling that will allow agents to be refuelled by either neutral, friendly or enemy agents or a combination of any of them. There will also be additional trigger states allow personality changes when an agent is refuelled. Additional terrain colours will be introduced to provide different terrain types. These are to include hills and light or dense bush. When an agent is on one of these terrain types its sensor and movement ranges will be modified as will the stealth parameter. A line of sight calculator is also planned which, when enabled, will prevent agents behind impassable terrain being shot at. Aside from the possible use of MHPCC for additional data collection as mentioned in Section 2.3.2 the multiple run mode will allow for step by step data to be recorded along with casualty locations.

## 2.4 Socrates

Socrates [13] began under Project Albert and evolved from a demonstration simulation called Simulation of Intelligent Reactive Entities (SIRE). It is coded in Java and being developed by Emergent Information Technologies, Inc.

Socrates (Version 2.1) is somewhat more difficult to use than the ABD's mentioned so far. There is currently no Windows environment to set up a scenario so input data must be edited via an XML file. Once an input file has been created it must be run via a command line in DOS. The scenario cannot be viewed while it is running, but can be viewed by the Socrates Viewer (Figure 13) once the run is completed. Run time for Socrates is also significantly higher than for EINSTEIN and MANA. This fact combined with the inability to view "live" runs can make setting up a scenario, and then refining it, a tedious process.

Movement in Socrates is significantly different to that of other ABD's as there are no distinct attraction and repulsion weights to the enemy or other friendlies. There are currently only two sides in Socrates, friendlies and enemies. Movement is largely based on an agent's user defined propensity to obey certain rules. These rules are hard coded and examples include staying within (or outside of) enemy sensor or firing range, to keep the commander in sensor range, to remain in formation (as specified by commander) and the amount of trust that you have in your commander.

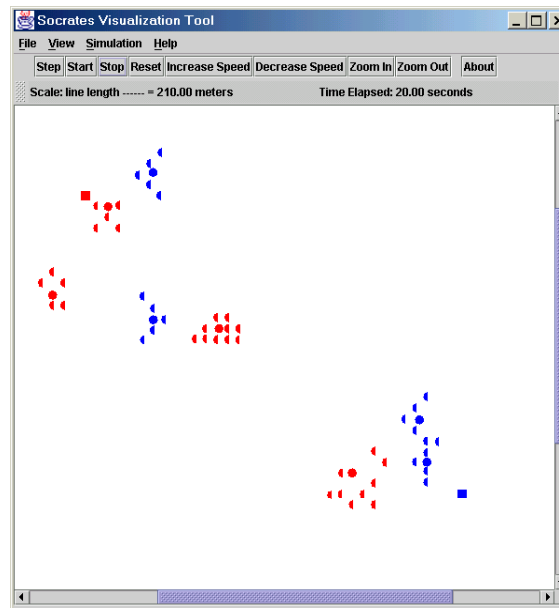


Figure 13: Socrates Viewer

Socrates also has a more detailed command structure than other ABD's. There are three command levels, usually defined by Grunts, Commanders and Leaders. Commanders make a decision on what formation to order the grunts into based on user defined tactical weights. These weights include the importance to remain evenly spaced, evade the enemy, attack the enemy, observe the enemy and not be surrounded

Agents in Socrates may also be given different values. Often Commanders and Leaders are given higher values than Grunts. Agents can then be given personalities to target high value agents. Commanders choose formations from a number of hard coded options, which include many traditional forms. Current formations include flank, line attack, line hold, line retreat and line tight or wide.

The only terrain currently modelled in Socrates is an obstruction that agents cannot pass through. Obstruction coordinates can be defined in the XML input file.

Like the ABD's mentioned already qualitative analysis can be done with Socrates although because of the significant time taken between setting up the scenario, running it, refining it and re-running it the analysis that can be performed is somewhat limited. The best method for data collection in Socrates is via a web submittal form to MHPCC. Like ISAAC up to five parameter spaces can be searched and the results can be viewed using the Albert VisTool. Socrates also has introduced some different measures of effectiveness along with the more traditional ones. They include the number of red or blue losses, the time taken for 100, 75, 50 and 25 per cent of red or blue to be lost, the time the first red or blue loss occurs and the number of decisions made by the red or blue leader.

One of the major difficulties with data collection and Socrates is the run time required. Although up to five parameters can be selected this is often infeasible. It is also hard to ensure that the results are statistically significant as it is often not feasible to run more than about 50 runs for each sample. Previous studies in MANA [9] have shown that some scenarios require up to 600 runs before settling down to a mean result.

Like MANA, Socrates is being developed continuously and many new features are being planned. Planned new features for the next release include allowing agents accept more risk to go after a high value target. Some code changes will allow run time to be reduced by a factor of two. This will improve data collection capabilities but will still not increase the run time speed to that of MANA or EINSTEIN. Up to eight sides will be able to be defined, each with a sentiment value between -1 and 1. This value represents how they view other agents with -1 representing much hatred and 1 representing much favour. Agents will also be able to store a reference of what he thinks the other agent's capabilities (ranges and Pk) are. Currently agents assume that other agents have the same capabilities as them.

## 2.5 Archimedes and Pythagoras

Archimedes [14] is an ABD that was funded by the USMC but is now being redeveloped by a different company and has been renamed Pythagoras. There has not yet been a release of Pythagoras but I will make a few comments on my experience with Archimedes.

Archimedes, like Socrates, has no specific attraction and repulsion weights to other agents. Instead the user must define a set of variables along with connections between these variables and the agents. Rules can then be defined to determine how agents react to certain information. Variables can be either fuzzy or crisp which allows for rules like the following:

If Player Range < far  
Set Movement Strength to strong.  
If Player Range is far enough  
Set Movement Strength to none.

These rules allow greater freedom when assigning behaviours to agents however it is not easy to implement and set up variables and agents, a task which is made even harder by the lack of any documentation on Archimedes.

Archimedes does have a very good terrain editor and terrain can modify agent's movement, visibility and stealth. Unfortunately due to problems constructing even the simplest of scenarios I have not been able to put these terrain features to full use.

When Archimedes was released there was a facility available to submit files to MHPCC as for ISAAC and Socrates however this has now been disabled while Pythagoras is being

developed. It is anticipated that Pythagoras will have a web based submittal facility to support extensive parameter explorations.

## 2.6 Bactowars

Bactowars [15] is an ABD being developed at Land Operations Division, DSTO to specifically focus on force mix scenarios in the littoral environment. Like Socrates, it is also written in JAVA and seems to suffer the same problems with regard to slow run times.

Bactowars is probably most similar to ISAAC in terms of agents and their behaviours. However, there are two types of agents in Bactowars, one is a physical agent like those in ISAAC, and the other is called a marker (or meme) agent which represent the ebb and flow of “the war of ideas”. These meme entities interact with other entities and can be likened to the influence or allegiance a side may have over the general population. Physical agents have the same general properties as those in ISAAC (sensor, firing, movement ranges etc) as well as a chance of spawning each other agent type (in most cases this will be zero). Each agent also has a chance of changing its type into each other agent type (in most cases this will be zero). Bactowars also has stealth and fuel parameters similar to those in MANA. A screen shot of the current Bactowars simulation screen is shown in Figure 14.

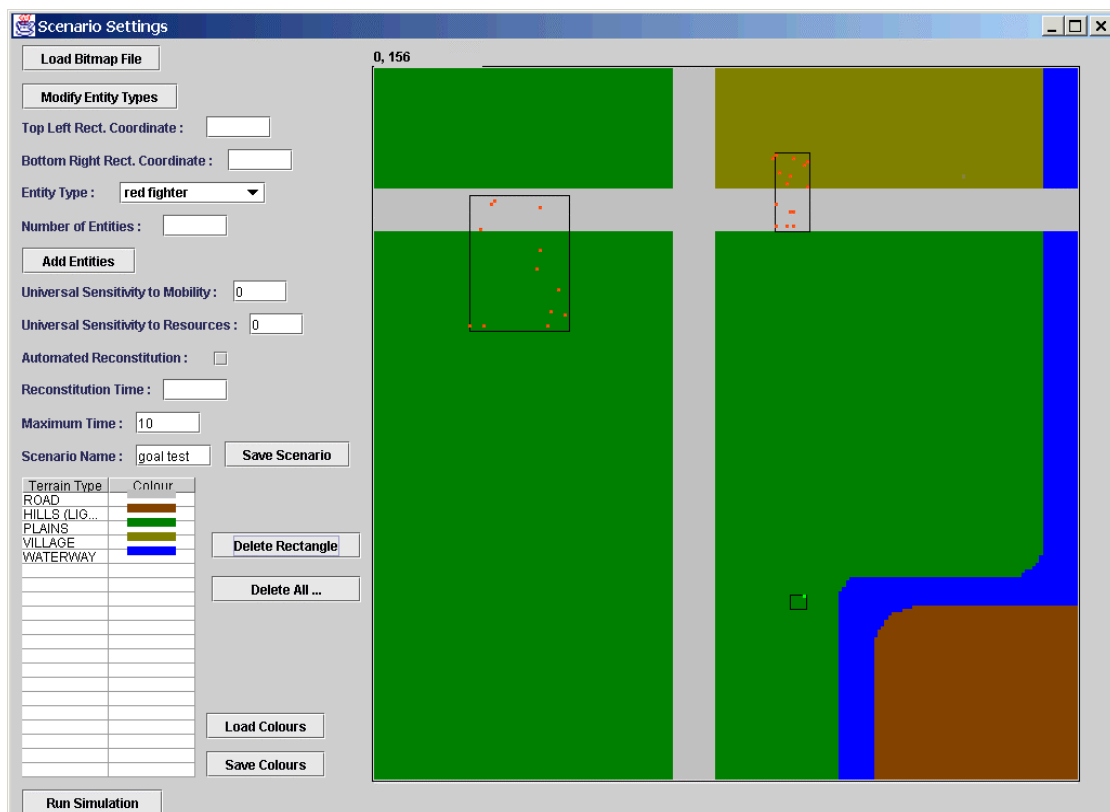


Figure 14: Bactowars scenario screen

Terrain in Bactowars can be created in any bitmap editor. The user can then define any different colours to represent different types of terrain simply by clicking on it and giving it a name. Currently the user can only modify an agent's movement for each type of terrain.

The only type of data collection available in Bactowars is qualitative analysis via interactive playback. It is hoped that a joint effort between DSTO, the Australian Army and the USMC may result in software being written so that results from Bactowars can be viewed using the Albert VisTool.

## 2.7 Comparison of the Current Suite of Agent Based Distillations

Table 1 summarises the different features of the ABD's listed above. As Archimedes is no longer being developed and Pythagoras has not yet been released they have not been included. The table is intended as a summary only and is not to suggest which ABD is superior. In the author's opinion no one model is suitable for all purposes. This table may be a guide to help select which model is most suitable for a certain scenario or question.

Table 1: Comparison of ABDs

	ISAAC	EINStein	MANA	Socrates	Bactowars
<b>Run Time</b>	Fast	Fast	Fast	Slow	Slow
<b>Ease to Set up</b>	Moderate, input via text editor	Fast, user friendly GUI	Fast, user friendly GUI	Slow, input via editing XML files	Fast, user friendly GUI
<b>Effect of Terrain</b>	Impassable objects only	User defined terrain affecting most parameters	Impassable objects and easy going terrain	Impassable objects only	User defined terrain affecting movement
<b>Stand Alone Multiple Time Series Mode</b>	No	Yes	Yes	No	Yes
<b>Automated Parameter Space Exploration</b>	Yes, only via MHPCC	Yes	No	Yes, only via MHPCC	No
<b>Documentation</b>	Yes	Yes	Yes	Yes	No
<b>Project Status</b>	Completed	Waiting for final release	Under Development, Version 2 due early 2002	Under Development, updated version due early 2002	Early stages of Development
<b>Personality Changes</b>	Only from alive to injured	Only from alive to injured	Multiple Trigger States	No, but tactics may change	Planned but not yet implemented
<b>Command Structure</b>	Three tier	Three tier	Two tier (still under development)	Three tier	No

### 3. Movement Within ABDs

The User Manual of MANA [12] states that “the most important action of an agent is to move.” This appears justified since being deliberately low-resolution means that the detailed physics of combat are largely ignored (or abstracted to simple constructs) and thus any interesting behaviour should appear as a result of the manoeuvring of the agents about the battlefield. Movement of agents within EINSTEIN and MANA ABD’s is based on a simple attraction-repulsion weighting system and an associated numerical penalty function. From its current location, the agent moves to the location within its movement range that incurs the least penalty. That is, the agent attempts to satisfy its personality-driven desire to move closer to or further away from other agents and either of the two flags. This algorithm is applied to each agent on both sides and each is moved to its new location. This process is repeated for each time step in the simulation.

The form of the penalty function implemented by both the EINSTEIN and MANA ABD’s is hard-coded. The user only has control over the value of the weightings towards the agents and flags. The user defines these weightings when a scenario is constructed and is chosen to represent surrogates for the tactics employed by the entities. For example, the EINSTEIN User manual [11] provides examples of aggressive (defensive) postures by assigning relatively large positive (negative) weights to enemy agents.

Given the simplistic nature of the attraction-repulsion weighting system, this will be at best an approximation to the true behaviour being modelled. What is important then, is that the movement algorithm implements faithfully the relationships that the user intends by assigning values to those weights. For example, consider the situation with weightings of +40 towards allied entities, -10 towards enemy entities, and +20 towards the enemy flag. The most natural interpretation of this situation is that the entity is four times more likely to move towards other allied entities than it is to move away from enemy entities, but that it is only two times more likely to move towards other allied entities than it is to move towards the enemy flag.

However, there are two key factors which are not stated in the above interpretation that are important in terms of what the user believes is being modelled. These factors are the number of entities the agent is aware of (generally those within its sensor range) and the distances those entities are from the agent in question. For example, does the above weighting system interpret the total weight for five enemy entities as -50, or is it independent of the number of enemy entities observed (or is it some non-linear function)? Similarly, does the above weighting system degrade the weight towards enemy entities as their distance from the agent in question increases (and is this degradation a linear function), or is it independent of these distances? Both of these questions are important for any weighting system in general, but are particularly relevant for those that reside in EINSTEIN and MANA.

The relative simplicity of ABD's, in particular EINStein and MANA, seems to have led to the general acceptance of their associated movement algorithms, without apparently much validation or analysis of behaviour. However, counter-intuitive results from a recent study using the EINStein ABD [16] suggest that the behaviours produced by its movement algorithm may not always be desirable or indeed what the user intended.

This section is based on a recent study [17] and examines in detail the movement algorithms of two of the more popular ABD's, EINStein and MANA. A one-dimensional scenario is examined in detail to illustrate the movement algorithms and to discover the causes of the unwanted behaviour. An alternative movement algorithm is then presented as a potential remedy and assessed. The new algorithm is a natural modification to both the EINStein and MANA movement algorithms. A suggestion for probabilistic movement selection, which draws heavily from simulated annealing, is also proposed as a future enhancement.

### 3.1 EINStein and MANA Movement Algorithms

#### 3.1.1 One-Dimensional Scenario Analysis

To illustrate the counter-intuitive behaviour than EINStein and MANA can produce, a simple scenario is presented in Figure 15 below to allow the penalty calculations to be explicitly performed. The single Blue agent is repelled by the Red agents (with weighting -10) and attracted to the flag (with weighting +20). Thus, there is a tension between wanting to move forwards towards the flag and remaining in place to avoid the Red agents. The decision that the Blue agent must make is whether to stay in place (remain at location 1) or move forwards (move to location 2).

While this scenario is not general (for example, having at least two Blue agents or allowing moves backwards) it is sufficient to illustrate the behaviours of MANA and EINStein while also allowing the threshold analysis in the next section to be performed analytically.





										
1	2	3	4	5	6	7	8	9	10	11

Figure 15: *Simplified One-Dimensional Scenario*

We examine this scenario with two variants, one where Blue has a sensor range of three and one where Blue has a sensor range of six. Each calculation below shows the enemy and flag components separately, followed by the total penalty. The equations for the two algorithms are also given although restricted to the parts relevant to this scenario (the enemy and flag components). The component for allies (other Blue agents from Blue's

perspective) is therefore omitted, but is identical in nature to that of the enemy component.

The equation that EINSTEIN uses to compute the penalty at each potential new location is given by

$$Z_{new} = \left( \frac{W_E}{E * R_S \sqrt{2}} \sum_{i=1}^E D_{i, new} \right) + W_F \left( \frac{D_{F, new}}{D_{F, old}} \right) \quad (1)$$

while the equation that MANA uses is given by

$$Z_{new} = \left( \frac{W_E}{100 * E} \right) \left( \sum_{i=1}^E \frac{D_{i, new} + (100 - D_{i, old})}{100} \right) + \left( \frac{W_F}{100} \right) \left( \frac{D_{F, new} + (100 - D_{F, old})}{100} \right). \quad (2)$$

Table 2 lists the definitions of each of the variables in these equations. Note that if E is zero (ie no entities defeated) then the first term is to be taken to be zero. Table 3 tabulates the two components and the total penalty for the choices (assuming a movement range of one square) of staying in place (location 1) or moving forwards (location 2), for the cases where the sensor range is either 3 or 6, for these different penalty functions.

Table 2: Penalty function variables

Variable	Definition
$R_S$	Sensor range of agent about to move
$E$	Number of enemy entities within sensor range
$W_E$	Weighting towards enemy agents
$D_{i, new}$	Distance to the $i$ th enemy from the new location
$D_{i, old}$	Distance to the $i$ th enemy from the current (old) location
$W_F$	Weighting towards the flag
$D_{F, new}$	Distance to the flag from the new location
$D_{F, old}$	Distance to the flag from the current (old) location

Table 3: Penalty function component calculations

Sensor Range	Location	Enemy Component	Flag Component	Penalty
<b>EINSTEIN Penalty Function</b>				
$R_S = 3$	Current (1)	-7.07	20.00	12.93
	New (2)	-4.71	18.00	13.29

$R_S = 6$	Current (1)	-5.30	20.00	14.70
	New (2)	-4.12	18.00	13.88
<b>MANA Penalty Function</b>				
$R_S = 3$	Current (1)	-0.10	0.20	0.10
	New (2)	-0.099	0.198	0.099
$R_S = 6$	Current (1)	-0.10	0.20	0.10
	New (2)	-0.099	0.198	0.099

For the EINSTEIN penalty function, where the sensor range is 3 (thus can only see the first Red agent), the Blue agent would decide to stay in place (since the penalty function is being minimized). However, with a sensor range of 6 (and can thus see both Red agents), the Blue agent apparently decides to move forward. This behaviour is counter-intuitive. For the MANA penalty function, for both cases (sensor range of 3 or 6) the Blue agent decides to move forward. However, we note that the penalties do not change at all when the Blue agent can see both Red agents. This behaviour is also counter-intuitive.

### 3.1.2 Discussion of Behaviour

Both algorithms above use scale factors in their penalty calculations. EINSTEIN uses the relative distance to the flag  $D_{F,new} / D_{F,old}$  but then scales the average distance to other entities (allies or enemy) by  $R_S\sqrt{2}$ . It is not clear to the author the rationale for this choice of scaling. It does, to an extent, scale the new distances relative to the old distances and to the relative distances used for the flag. However, as seen above it doesn't solve the problem entirely, and as will be seen below there is a simpler solution. EINSTEIN also scales the summation term by the number of entities (that is, it calculates the average distance). In effect, this implies that the Blue agent only effectively observes one Red entity when deciding on which move to make (and this one entity is positioned at the centroid of the Red entities within sensor range).

These observations explain why the Blue agent remains in place for a sensor range of 3 (because it "sees" one Red entity 3 units away) but moves forward for a sensor range of 6 (because it "sees" only one Red entity 4.5 units away, when in fact there are two Red entities at 3 and 6 units away respectively). This again seems to be undesirable behaviour and not what the user would have intended when the original weights were entered.

MANA does not use relative distance scaling for the flag or other entities in the fashion used by EINSTEIN. Instead, it scales all distances by 100. That is, it treats all entities as if they were 100 units away. This means that the penalty for moving towards a Red that is 5 units away will be the same as that for moving towards a Red that is 50 units away. Again, it appears to the author that this choice is both too arbitrary and unnecessary. MANA also uses the average distance concept and thus also falls victim to the 'centroid' problem.

It is clear from this simple example that there are two factors in the penalty functions used by EINSTEIN and MANA which are fundamental in terms of implementing what the user

believes is being modelled when attraction-repulsion weights are prescribed. These factors are the number of entities the agent is aware of and the distances those entities are from the agent.

## 3.2 Alternative Movement Algorithm

### 3.2.1 Threshold Analysis

In the simple one-dimensional scenario above with a sensor range of three, Blue will choose to move forward if the absolute difference in the flag components between  $Z_1$  (stay in place) and  $Z_2$  (move forward) is greater than the absolute difference in the enemy components between  $Z_1$  and  $Z_2$ . For the Einstein case, these two quantities are:

$$\Delta Z_{Flag} = \frac{W_F * D_F}{D_F} - \frac{W_F(D_F - 1)}{D_F} = \frac{W_F}{D_F} \quad (3)$$

and

$$\Delta Z_{Enemy} = \frac{W_E * (D_E - 1)}{R_S \sqrt{2}} - \frac{W_E * D_E}{R_S \sqrt{2}} = -\frac{W_E}{R_S \sqrt{2}} \quad (4)$$

where  $D_F$  and  $D_E$  are the distances to the flag and to the Red entity from the original location, respectively. Thus, Blue will choose to move forward in Einstein when:

$$\frac{W_F}{D_F} > -\frac{W_E}{R_S \sqrt{2}} \Rightarrow -\frac{W_F}{W_E} > \frac{D_F}{R_S \sqrt{2}}. \quad (5)$$

The left hand side of this inequality implies that the weights obey a proportional or relative law. That is, the ratio of the weights must exceed a certain threshold in order to affect a move forward. This threshold is the left hand side of the inequality and the point to note is that it is independent of  $D_E$  the distance to the enemy, but does depend on the unusual scaling factor discussed above.

A similar analysis for the MANA penalty function yields:

$$\Delta Z_{Flag} = \left( \frac{W_F}{100} \right) \left( \frac{D_F + (100 - D_F)}{100} \right) - \left( \frac{W_F}{100} \right) \left( \frac{D_F + (100 - D_F - 1)}{100} \right) = \frac{W_F}{100^2} \quad (6)$$

and

$$\Delta Z_{Enemy} = \left( \frac{W_E}{100} \right) \left( \frac{D_E + (100 - D_E - 1)}{100} \right) - \left( \frac{W_E}{100} \right) \left( \frac{D_E + (100 - D_E)}{100} \right) = -\frac{W_E}{100^2} \quad (7)$$

Thus, Blue will choose to move forward in MANA when:

$$\frac{W_F}{100^2} > -\frac{W_E}{100^2} \Rightarrow -\frac{W_F}{W_E} > 1, \quad (8)$$

where we note that the threshold for movement here is independent of distance altogether (either  $D_E$  or  $D_F$ ). We argue that a more natural threshold for movement is given by:

$$-\frac{W_F}{W_E} > \left(\frac{D_F}{D_E}\right)^r \quad (r \geq 0) \quad (9)$$

which implies that Blue will move forward provided the attraction towards the flag exceeds the repulsion from the enemy by more than the distance to the flag exceeds the distance to the enemy, scaled by a power (equally, (9) can be interpreted in terms of a threshold condition for the ratio of the distances, since these vary during the scenario while the weights are fixed).

The value of  $r$  controls how important the relative distance between the flag and the enemy is. As  $r$  tends to zero, all entities are considered to be the same distance away and the local average is used (similar to MANA). As  $r$  increases the reduction in weight towards entities further away becomes more and more significant. Note that (9) is only appropriate when there is some form of tension (in this case, being attracted to the flag but repelled from the enemy).

### 3.2.2 Inverse Distance Estimators

The idea of modifying the distance by some power is also common to techniques for spatial interpolation [18] which use the approximation

$$z(\mathbf{x}_0) \approx z^*(\mathbf{x}_0) = \sum_{i=1}^N w_i z(\mathbf{x}_i). \quad (10)$$

Here  $z^*$  is the estimate of the true value  $z$  at a specific location  $\mathbf{x}_0$ ;  $z(\mathbf{x}_i)$ ,  $i = 1, \dots, N$  are  $N$  nearby sample values at locations  $\mathbf{x}_i$ , and  $w_i$  are the  $N$  weights to be chosen. Inverse distance estimators date back to the 1920's and are still frequently used and form part of most commercial contouring packages. These techniques give greater weight to closer samples by assigning

$$w_i = K \|\mathbf{x}_i - \mathbf{x}_0\|^{-r}, \quad i = 1, \dots, N \quad (11)$$

where  $\|\cdot\|$  is a distance norm chosen to account for possible anisotropies,  $r$  is a non-negative parameter chosen to reflect the degree of spatial continuity, and  $K$  is a normalizing constant such that the weights sum to one.

To investigate the use of the concepts underlying inverse distance estimators to produce a penalty function for a movement algorithm, we replaced the values  $z(\mathbf{x}_i)$  with the weightings assigned to detected ally and enemy entities, and the flag. Then, by applying

the inverse distance estimator to approximate the variable  $z$  at each of the possible new locations produces numerical values that can be viewed as penalty values. The best move in each case would be to the location with the highest penalty (as opposed to the more usual minimization).

The penalty function based on inverse distance estimation is then given by:

$$Z_{new} = \left( \frac{W_E}{E^\alpha} \sum_{i=1}^E \frac{1}{D_{i,new}^r} \right) + \left( \frac{W_A}{A^\alpha} \sum_{j=1}^A \frac{1}{D_{j,new}^r} \right) + \frac{W_F}{D_{F,new}^r} \quad (12)$$

where  $A$ ,  $W_A$  and  $D_{j,new}$  are the ally analogues to those defined in Table 2 for the enemy penalty component, and where  $r$  is a user-specified non-negative rate factor. The idea with this technique is that agents that are further away are given less weight. As  $r$  tends to zero, all entities are considered to be the same distance away and the local average is used (similar to MANA). As  $r$  increases the reduction in weight towards entities further away becomes more and more significant. Coincidentally, we discovered that a similar approach had been used in a social science cellular automata model some thirty years earlier [5].

However, when performing the threshold analysis for the one-dimensional scenario on the inverse distance based penalty function, we get

$$\Delta Z_{Flag} = \frac{W_F}{D_F^r} - \frac{W_F}{(D_F - 1)^r} = W_F \left( \frac{(D_F - 1)^r - D_F^r}{D_F^r (D_F - 1)^r} \right) \quad (13)$$

and

$$\Delta Z_{Enemy} = \frac{W_E}{(D_E - 1)^r} - \frac{W_E}{D_E^r} = -W_E \left( \frac{(D_E - 1)^r - D_E^r}{D_E^r (D_E - 1)^r} \right). \quad (14)$$

So Blue will choose to move forward when

$$\frac{W_F}{W_E} > - \left( \frac{D_F}{D_E} \right)^r \left( \frac{D_F - 1}{D_E - 1} \right)^r \left( \frac{(D_E - 1)^r - D_E^r}{(D_F - 1)^r - D_F^r} \right). \quad (15)$$

Using the infinite series binomial series expansion (convergent since  $1/D_E < 1$  and  $1/D_F < 1$ ) we get

$$LHS = \left( \frac{D_F - 1}{D_E - 1} \right)^r \left( \frac{\frac{-r}{D_E} + \frac{1}{D_E^2} - \frac{1}{D_E^3} + \dots}{\frac{-r}{D_F} + \frac{1}{D_F^2} - \frac{1}{D_F^3} + \dots} \right). \quad (16)$$

Assuming  $D_F$  and  $D_E$  are much larger than unity, we can approximate  $D_F-1$  with  $D_F$  (similarly for  $D_E-1$ ), and approximate both infinite series with their leading terms only. This allows the threshold to be simplified and the approximate result is

$$-\frac{W_F}{W_E} > \left( \frac{D_F}{D_E} \right)^{r+1}. \quad (17)$$

This is similar to the desired threshold given by (9) but since the parameter  $r$  is non-negative it is not as general (as will be illustrated below).

### 3.2.3 Relative Distance Estimators

An alternative movement algorithm, which can be shown to produce the desired movement threshold given in (9), is hereby proposed whereby the denominator of the enemy (and ally) components of the penalty are divided (or scaled) by  $D_{i,old}$  and  $D_{j,new}$  (the distances from the current location to the location of the  $i$ -th enemy and  $j$ -th ally, respectively). This would replace the artificial and arbitrary scaling factors of EINSTEIN and MANA (sensor range and the constant 100). The new function also allows the exponent to take on values between zero and one, unlike the inverse distance equation.

The alternative penalty function is then given by

$$Z_{new} = \left( \frac{W_E}{E} \right) \sum_{i=1}^E \left( \frac{D_{i,new} - D_{i,old}}{D_{i,old}} \right)^r + \left( \frac{W_A}{A} \right) \sum_{j=1}^A \left( \frac{D_{j,new} - D_{j,old}}{D_{j,old}} \right)^r + W_F \left( \frac{D_{F,new} - D_{F,old}}{D_{F,old}} \right)^r \quad (18)$$

Using the difference of distances in the numerators and the old distances in the denominator means that the change in distances (from the current location to the new location) are compared, and compared in a relative sense. Note that the power of  $r$  should be implemented such that if the argument is negative then the result should be the negative of the  $r$ -th power of the absolute value of the argument. This is to ensure that the effect of moving closer or further away is not lost when the power of  $r$  is applied. An added bonus is that the penalty for staying in place is by definition equal to zero and thus needn't be calculated (thus saving some computation time).

The use of relative distances in this fashion appears to solve the scaling problem inherent in both EINSTEIN and MANA. Under this new penalty function, entities would then assign a stronger weight to those entities that are nearby than that to those far away. For example, in the simplified one-dimensional scenario above, when the Blue entity moves forward it is 33% closer to the first Red entity, only 17% closer to the second Red entity, and only 10% closer to the flag, and these relative percentages will be reflected under the new penalty function.

If we now perform the threshold analysis on (18) we obtain

$$\Delta Z_{Flag} = W_F \left( \frac{D_F - D_F}{D_F} \right)^r - W_F \left( \frac{D_F - 1 - D_F}{D_F} \right)^r = \frac{W_F}{D_F^r} \quad (20)$$

and

$$\Delta Z_{Enemy} = W_E \left( \frac{D_E - 1 - D_E}{D_E} \right)^r - W_E \left( \frac{D_E - D_E}{D_E} \right)^r = -\frac{W_E}{D_E^r} \quad (21)$$

so Blue will choose to move when

$$\frac{W_F}{D_F^r} > -\frac{W_E}{D_E^r} \Rightarrow -\frac{W_F}{W_E} > \left( \frac{D_F}{D_E} \right)^r \quad (22)$$

which produces the desired threshold (noting however that this is a general result only for the simplified situation given in Figure 15).

### 3.2.4 Utility Curves

To illustrate the flexibility of this new movement algorithm, and of the parameter  $r$ , it is useful to consider the penalty function as a form of utility curve. Currently the penalty functions of both EINSTEIN and MANA essentially implement a linear utility curve. That is, if we denote  $X = D_{i,j}/D_{i,0}$  as the relative distance of the new location from the  $i$ -th entity to the current location from the  $i$ -th entity, then the penalty functions are based on the utility curve  $W^*(X-1)$ , where  $W$  is the weighting towards the  $i$ -th entity type.

However, this implies that to obtain 'full positive' utility (in the sense of receiving all of  $W$ ) then the new location must be such that the relative distance is doubled ( $D_{i,j} = 2 * D_{i,0}$  and thus  $X=2$ ). Similarly, it also implies that to obtain 'full negative' utility (in the sense of receiving all of  $-W$ ) then the new location must be such that the relative distance is zero ( $D_{i,j}=0$  and thus  $X=0$ ).

It appears to the author that there may be situations where this change in utility would not be linear. For example, it sounds equally intuitive that full positive and full negative utility be assigned if the relative distances were say doubled and halved, respectively. This would require some form of non-linear utility curve. The new penalty function above provide flexibility in defining the utility curve through the parameter  $r$ . Figure 16 displays the form of the utility curve for various values of  $r$  (with  $W$  set to unity for simplicity). We see that for  $r=1$  the utility curve is linear, as in EINSTEIN and MANA.

However, the figure also shows that by varying the value of  $r$  the user can obtain non-linear utility curves. For example, the case  $r=0.1$  allows full positive and negative weights to be assigned if the relative distance is increased, or decreased, by only a small amount, respectively (or 'faster than linear'). Similarly, for values of  $r$  greater than one 'slower than linear' behaviour can be modelled, which means that relatively large changes in distance is required to produce a significant weighting. The author believe that this flexibility may

assist the user when determining exactly how the personality weightings are to be modelled and interpreted.

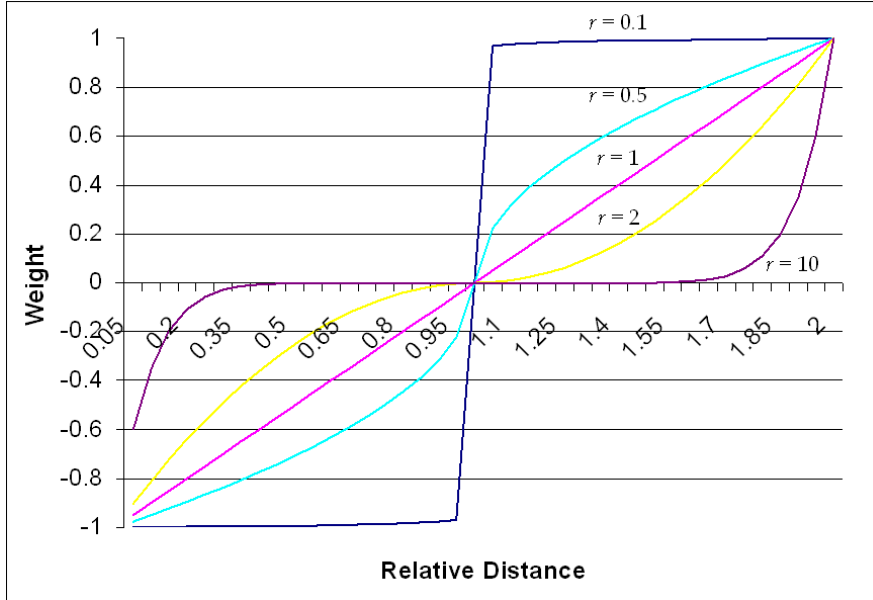


Figure 16: Utility Curves Generated by the New Penalty Function

It should be pointed out that Socrates [13] also implements a non-linear utility curve, however its movement diagnostics are not as straight forward as the simple attraction-repulsion weighting system used in EINSTEIN and MANA which has been the focus of this paper, and thus remains as a future research area.

### 3.2.5 Cumulative Penalties

As presented above, the new penalty function still computes an average (relative) distance as it divides by the number of enemy within sensor range, and is thus subject to the ‘centroid’ issue discussed previously. It appears to the author that a more usual form of the penalty calculation would use a cumulative function instead of an average. This would involve simply removing the denominator at the front of the previous equation. However, at times it may not be desirable to use either the cumulative or average functional. A generalisation of the above penalty function to incorporate this is given simply by

$$Z_{new} = \left( \frac{W_E}{E^\alpha} \right) \sum_{i=1}^E \left( \frac{D_{i,new} - D_{i,old}}{D_{i,old}} \right)^r + \left( \frac{W_A}{A^\alpha} \right) \sum_{j=1}^A \left( \frac{D_{j,new} - D_{j,old}}{D_{j,old}} \right)^r + W_F \left( \frac{D_{F,new} - D_{F,old}}{D_{F,old}} \right)^r \quad (23)$$

where  $\alpha$  is a real value between zero and one.

### 3.2.6 One Dimensional Scenario Analysis

Equation (23) represents our proposed improved movement algorithm penalty function, and is defined by two user-selectable parameters  $\alpha$  and  $r$ . The simplified one-dimensional scenario is now re-examined. Three values of  $\alpha$  (0, 0.5 and 1) and  $r$  (0.5, 1 and 2) were used and the results are summarised in Table 4.

Table 4: New Penalty Function Results on the Simplified One-Dimensional Scenario

$r$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
<b>0.5</b>	MS	MS	MM
<b>1.0</b>	SS	SS	SS
<b>2.0</b>	SS	SS	SS

There are four possible combinations that could occur for the two cases of sensor range of 3 and sensor range of 6, respectively. The Stay then Move combination (listed as SM) is the only one that is illogical. The Move then Stay (MS) possibility is logical while the other two (SS or MM), where the entity either stays or moves regardless of the sensor range, are also both feasible depending on the interpretation of the weighing system by the user. Recall that EINSTEIN produced the SM behaviour, while MANA produced MM.

All of the results for the new equation are logical. The effect of  $\alpha$  can be clearly seen in the  $r=0.5$  example. When the penalty is entirely cumulative ( $\alpha=0$ ) the entity does not move when the extra enemy is seen. However when the penalty takes the average distance to the two enemies ( $\alpha=1$ ) the entity moves in both cases. This is similar to the MANA example earlier.

The effect of  $r$  is also significant. Table 5 lists the flag and enemy components of the penalty for the case where only one enemy is visible (sensor range = 3), making the result independent of the value of  $\alpha$ . This table shows that although the magnitude of both components decreases as  $r$  increases, the ratio of the magnitude of the enemy component to the flag component increases, i.e. more importance is placed on things that are closer (the enemy in this example) as  $r$  increases.

Table 5: Effect of the  $r$  parameter on penalty components

$r$	Flag Comp	Enemy Comp	Ratio  E:F
0.5	-6.324	5.773	0.913
1.0	-2.000	3.333	1.667
2.0	-0.200	1.111	5.555

### 3.2.7 Two-Dimensional Scenario Analysis

A set of trials was performed using a 24 by 24 grid consisting of 60 enemy entities and 30 ally entities randomly distributed. For simplicity, these entities remain stationary. A flag was located at position towards the top left corner and the start position of the entity we wish to track was on the eastern border. This entity is attracted to the flag (with weighting of +100) and allies (with weighting of +20) and repelled from the enemy (with weighting of -50) and has a sensor range of five. Table 6 and Figure 17 summarise the results for nine combinations of  $\alpha$  and  $r$ .

Table 6: *Relative distance movement algorithm paths*

$r$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
<b>0.5</b>	Path 2	Path 1	Path 1
<b>1.0</b>	Path 2	Path 2	Path 1
<b>2.0</b>	Path 2	Path 2	Path 2

Path 1 refers to the path initially travelling southwest in Figure 17 while Path 2 refers to the shorter path travelling south. Each combination of  $\alpha$  and  $r$  actually varied slightly from these paths but for clarity in Figure 17 was not displayed. The reason for the similarity in several paths is due to the static nature of this test, and would be more variable if tested in a dynamic ABD. Both MANA and EINSTEIN follow the path travelling northwest.

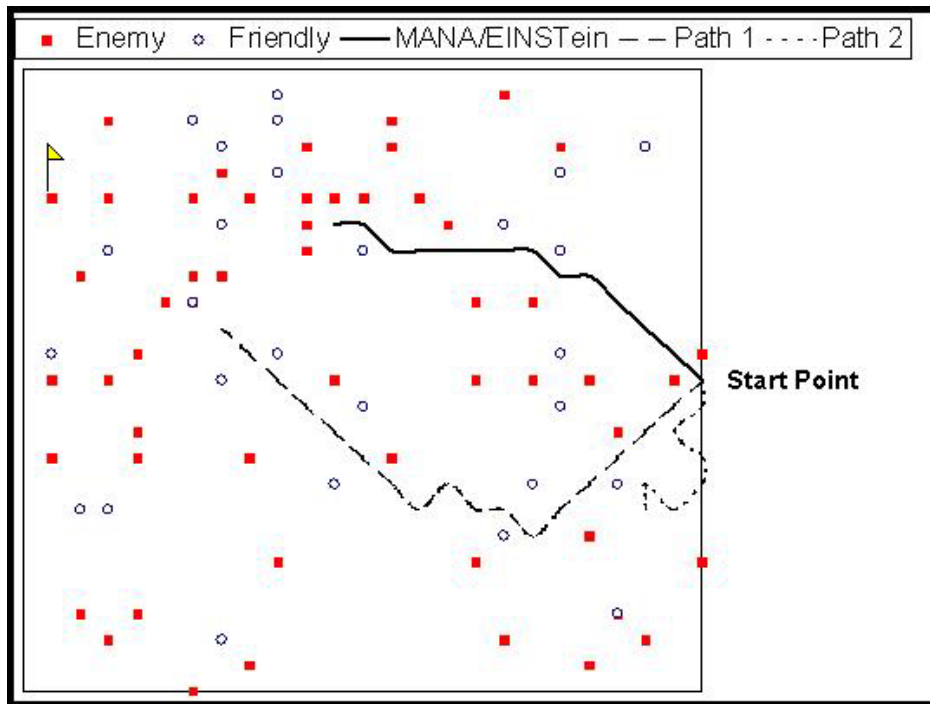


Figure 17: Variation in Paths Generated by Alternative Movement Algorithms

The first observation to make is the distinct difference in paths taken by MANA, EINSTEIN and the new movement algorithm, thereby supporting our view of the importance the movement algorithm can have to scenario results. Both MANA and EINSTEIN appear to have little concern in initially moving through the two enemy entities to the north and west, and end their movement very close to a wall of enemy. The new movement algorithm appears to take a more cautious approach.

Table 6 highlights the effect the two parameters have in this new movement algorithm. As  $\alpha$  increases the penalty closer resembles the average functional, as used by MANA and EINSTEIN. As a result the entity becomes more strongly attracted to the flag, as the other entities are no longer being counted cumulatively. However, as  $r$  increases the penalty function gives a much stronger weight to entities that are closest which means that the flag is given less weight. In the case of  $r=2$  the flag is virtually excluded altogether and only entities that are very close are given significant weighting. In a dynamic environment,  $r$  could become a function of tempo whereby high tempo would imply higher values of  $r$  and more attention is paid to closer enemy entities than in low tempo situations.

### 3.3 Extensions

#### 3.3.1 Stochastic Movement

The MANA model incorporates some form of randomness into its movement algorithm in an attempt to “notionally represent small differences in the personality, or ‘mood’, of the automata” and to “produce more realistic behaviour” [3, 9]. This is achieved by essentially limiting the number of decimal places kept in the penalty calculation, so that proportionally more ties are encountered and a random draw is used to select the new location.

An alternative method proposed here is to interpret the penalty (appropriately scaled, see below) at each new location as the probability of moving there. This appears to the author as a more natural and flexible approach, since all potential locations are candidates for the move selection but with appropriate relative frequencies. This approach has parallels with the technique of simulated annealing [19]. In both cases, it may be desirable for poorer solutions to be (temporarily) selected, in order to avoid being trapped in a local optimum and thereby increasing the chances of discovering a better solution on the next (or subsequent) penalty calculation. Specifically for combat ABD’s, the incorporation of stochastic movement could have applications for obstacle negotiation.

The need to balance exploitation (always choosing the best move) with exploration (choosing alternative moves) can lead to a number of move selection schemes. First, let us define  $Z_i$ ,  $i = 0, \dots, I$  as the penalty for moving to the  $i$ -th location. Then, the Greedy movement selection scheme chooses that location  $i^*$  such that  $Z_{i^*} = \min(Z_i)$ . This is the deterministic scheme used above and is designed to maximise exploitation. This scheme can be ‘relaxed’ to explore (randomly) with probability  $\varepsilon$  by setting

$$P(\text{Move} = i^*) = 1 - \varepsilon \text{ and } P(\text{Move} = i) = \frac{\varepsilon}{I}, i \neq i^*. \quad (24)$$

This rule is known as the  $\varepsilon$ -Greedy movement selection scheme and the parameter  $\varepsilon$  controls the balance between exploitation and exploration. However, the exploration is random so that the worst move is equally likely to be chosen as any other, which may not be desirable.

A better approach is to use the Soft-Max movement selection scheme [20], which explores with a graded probability distribution. The Greedy movement is still given the highest selection probability, but all others are weighted according to their respective penalties, and is defined by

$$P(\text{Move} = i) = \frac{\exp\left(\frac{-Z_i}{\tau}\right)}{\sum_{j=0}^I \exp\left(\frac{-Z_j}{\tau}\right)}, i = 0, \dots, I. \quad (25)$$

This is known as the Boltzman distribution and the parameter  $\tau$  is known as the temperature (from the annealing concept [10]) and is used to control the balance between exploitation and exploration. As  $\tau \rightarrow 0$  this scheme tends to the Greedy movement selection scheme (maximising exploitation) while  $\tau \rightarrow \infty$  produces the  $\epsilon$ -Greedy movement selection scheme with  $\epsilon=1/(I+1)$  which is equivalent to a uniform distribution (thereby maximising exploration).

To illustrate the differences between these schemes, consider the following situation. An agent is located at position 0 and has to choose between remaining in place or moving North, South, East or West. Suppose the corresponding penalties are  $Z_1 = -0.56$ ,  $Z_2 = -0.59$ ,  $Z_3 = -0.2$  and  $Z_4 = +0.1$ , respectively (while  $Z_0 = 0$  for remaining in place). Setting the control parameters to  $\epsilon = 0.3$  and  $\tau = 1$ , and the ‘movement precision’ of MANA to one decimal place, yields the probability distributions given in Table 7.

Table 7: Movement Selection Scheme Probability Distributions

Location	Penalty $Z_i$	Greedy	MANA	$\epsilon$ -Greedy	SoftMax
In Place	0.00	0	0	0.08	0.15
North	- 0.56	0	0.5	0.08	0.26
South	- 0.59	1	0.5	0.70	0.27
East	- 0.20	0	0	0.08	0.18
West	+ 0.10	0	0	0.08	0.14

The Greedy and MANA movement selection schemes are similar in that most moves are given zero probability, thus tending to focus on exploitation. Conversely, both the  $\epsilon$ -Greedy and SoftMax schemes allow all moves to be selected, however the  $\epsilon$ -Greedy scheme assigns the same probability of selection the worst move (West) as the second best move (North). The SoftMax scheme however, maintains the rankings observed by the penalties.

### 3.3.2 The Flag

For scenarios that involve a non-zero weighting towards the opponents flag (which generally represents the achievement of some form of objective), moving away from using an average functional ( $\alpha < 1$  in our new movement algorithm) implies that the flag component of the penalty will become increasingly insignificant. This was illustrated in the two-dimensional scenario above, which also showed that this effect could be mitigated to some degree by decreasing the value of  $r$ . However, if the user wishes the interpretation of the assigned weighting towards the flag to be more resistant to the effects of  $\alpha$ , then modifications are required.

Four forms for an alternative flag component of the penalty are listed below along with a description of what effect they are attempting to achieve. Note that  $A$  is defined as the number of allies, or Blue agents. The first alternative is the default case, as given in (23) in the main body of this report

$$Z_{F1} = S(D_{F, new} - D_{F, old})W_F \left| \frac{D_{F, new} - D_{F, old}}{D_{F, old}} \right|^r \quad (26)$$

and is given here explicitly only to allow comparison with the other alternatives. This form assumes that there is only one flag and the scaling factor is the distance to the current location to the flag. The second alternative is given by

$$Z_{F2} = S(D_{F, new} - D_{F, old})W_F \left| \frac{D_{F, new} - D_{F, old}}{\frac{1}{E} \sum_{i=1}^E D_{i, old} + \frac{1}{A} \sum_{i=1}^A D_{i, old}} \right|^r \quad (27)$$

which assumes that there is only one flag but uses a scaling factor which is the average distance of all detected enemy and ally entities. The effect of this in the simple scenario is that the flag is in essence now located between locations five and six, as illustrated in Figure 18.

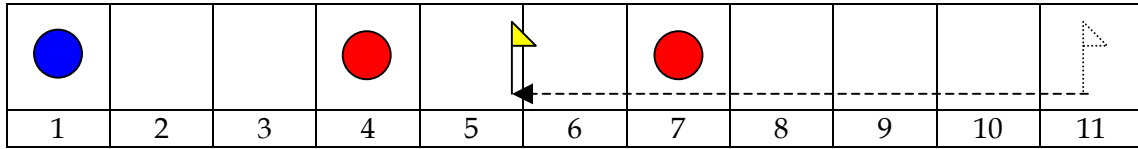


Figure 18: Effect of the Second Flag Alternative

The third alternative is given by

$$Z_{F3} = (E + A)^{1-\alpha} S(D_{F, new} - D_{F, old})W_F \left| \frac{D_{F, new} - D_{F, old}}{D_{F, old}} \right|^r \quad (28)$$

which assumes that there are as many flags as detected entities ( $E+A$ ). In the simple scenario there would now be multiple flags (2 in this case as there are only two other entities detected) positioned at location 11 as illustrated in Figure 19.





										
1	2	3	4	5	6	7	8	9	10	11

Figure 19: Effect of the Third Flag Alternative

The fourth alternative is given by

$$Z_{F4} = (E + A)^{1-\alpha} S(D_{F, new} - D_{F, old}) W_F \left| \frac{D_{F, new} - D_{F, old}}{\frac{1}{E} \sum_{i=1}^E D_{i, old} + \frac{1}{A} \sum_{i=1}^A D_{i, old}} \right|^r \quad (29)$$

which assumes that there are multiple flags and uses the scaling factor that is the average distance of all detected enemy and ally entities. In the simple scenario there would now be multiple flags (2 in this case) in essence now located between locations five and six as illustrated in Figure 20.






										
1	2	3	4	5	6	7	8	9	10	11

Figure 20: Effect of the Fourth Flag Alternative

Thus, in summary the proposed alternative penalty functions are given by

$$Z_{new} = \left( \frac{W_E}{E^\alpha} \right) \sum_{i=1}^E S(D_{i, new} - D_{i, old}) \left| \frac{D_{i, new} - D_{i, old}}{D_{i, old}} \right|^r + \left( \frac{W_A}{A^\alpha} \right) \sum_{i=1}^A S(D_{i, new} - D_{i, old}) \left| \frac{D_{i, new} - D_{i, old}}{D_{i, old}} \right|^r + Z_{Fj} \quad (30)$$

where  $Z_{Fj}$  is one of the flag component alternatives above ( $j=1, 2, 3, 4$ ).

Each of these alternatives appeals intuitively to the author to what is envisaged by the attraction-repulsion weighting system, however neither could confidently be claimed as 'correct' and the other 'incorrect'. The best approach would be to allow the user to select which alternative best represents the behaviour desired for the scenario under consideration.

### 3.4 Summary and Future Research

Based on counter-intuitive results from a recent study conducted by the author using the EINSTEIN ABD, the movement algorithms of that ABD and the MANA ABD were investigated. A simplified one-dimensional scenario was used to deduce the causes for the

counter-intuitive behaviour and some simple analysis led to a suggestion for improvement to the penalty function based on scaled relative distances and a partial-cumulative functional. The new movement algorithm is then defined by setting two parameters. Calculations show that this new movement algorithm does not suffer from the counter-intuitive behaviours that appear to question the validity of the EINSTEIN and MANA penalty functions and thus associated movement algorithms, and importantly, that significantly different paths can be generated.

Given the flexibility of the new movement algorithm, we suggest that a prudent approach is to make the penalty function a user-defined 'parameter', in the same way that the entity capabilities and personalities currently are. At a minimum, it would make more explicit the assumptions made about the movement algorithm when one provides the results of a study. Ideally, the robustness of these conclusions should be tested to variations in the movement algorithm, in the same way that sensitivity analysis is applied to other more traditional parameters.

An extension to any movement algorithm to incorporate stochastic movement was also proposed. Stochastic movement may assist in obstacle negotiation. The technique is based on selection schemes used in simulated annealing, which we argue is a more consistent scheme than that produced by the movement precision parameter of MANA.

Work that remains to be completed include the following. First, it is important to determine the combat utility of the new movement algorithm by using it within an ABD to evaluate a range of scenarios. More experiments need to be performed to determine the effect of the various flag penalty components on the behaviour of the agents. Early indications with the MANA ABD [21] appear to confirm its usefulness. Second, tests using the SoftMax movement selection scheme to produce stochastic movement need to be performed to explore its potential for combat situations as hypothesised above. Finally, the movement algorithms of other ABD need to be analysed in more detail.

## 4. Case Studies

### 4.1 Manoeuvre Operations in the Littoral Environment

#### 4.1.1 The Scenario

The Australian Army's Headline Experiment (HE) in 1999 was designed to provide information addressing the combat effectiveness of a 2015 Enhanced Combat Force. A central question was how Army's manoeuvre concepts might need to change. To experiment with ABD we abstracted a problem based on Manoeuvre Operations in a Littoral Environment (MOLE) and the specific hypothesis to be tested was whether a

small, mobile force with high situational awareness coupled with effective reach-back munitions could defeat a significantly larger force [16].

A three-day workshop investigated this proposition employing the EINSTEIN distillation to facilitate the study. The workshop had three aims. First, a number of baseline scenarios were to be constructed which modelled the units and mission as best could be achieved. As a result of this process, two subsequent aims should also have been achieved. They are, to determine some of the limits of applicability and resolution of the EINSTEIN distillation in modelling or representing Army capabilities and missions, and to develop within the Combined Arms Training Development Centre (CATDC)-DSTO group an increased level of proficiency in the use of ABD.

#### 4.1.1.1 Parameter Specification

The main physical characteristics of each element are presented in Table 8. The Blue force consists of a mix of light armoured vehicles (LAV), armed reconnaissance helicopters (ARH) and high-mobility artillery rocket system (HIMARS). For the baseline scenario, the force mix is such that there are 10 LAV, five ARH and one HIMARS unit, while the Red force consists entirely of tanks (45 T-80's). Thus, the Red-to-Blue force ratio is approximately 3:1. Note that the use of the terms LAV, HIMARS etc, are for convenience only, since the level of model resolution within ABD's implies that one is in fact employing surrogates (simplified abstractions) for the actual combat capabilities.

The LAV have relatively good speed and sensor range, but relatively poorer weapon characteristics. The task for the LAV is to survey the likely approaches of the enemy and to communicate detections back to the ARH and HIMARS units for prosecution. The ARH are significantly faster than the LAV and have double their sensor and weapon performance, however there are fewer of these assets. The task for the ARH is to quickly move to the location of detected enemy and decisively engage, based on the communicated information supplied by the LAV. The HIMARS unit is a single asset held at the rear of operations and brings heavy, lethal area-fire onto regions of detected enemy supplied by the LAV. The T-80 has half the movement and sensor characteristics of the opposing LAV, but have double the weapon performance and out-number the LAV.

Table 8: Major physical characteristics

Characteristic	LAV	ARH	HIMARS	T-80
Speed	2	4	4	1
Sensor	4	8	5	2
Fire	2	4	5	2
Lethality	0.25	0.5	0.4	0.5
Number	10	5	1	45

To simulate reconnaissance behaviour, 'negative attractiveness' to friendly and enemy entities is used. The former is used to create a dispersed reconnaissance force, while the latter is used to ensure the LAV does not become decisively engaged. A high attractiveness to the Area entity is used to simulate an area of operations (AO) assigned to the LAV force. The Cluster 'meta-personality' was also used to further enhance the dispersed nature of the LAV force. Similar entity definitions can then be constructed for the other units (ARH, HIMARS, T-80) to simulate the required characteristics and behaviours.

#### 4.1.1.2 HIMARS Modelling

HIMARS proved the most difficult entity to represent. EINSTEIN does not explicitly model indirect or area-fire weapons, in particular the forward observer concept. The closest approximation was to assign a grenade to a HIMARS squad that was given a low sensor range and a high movement range to allow it to quickly react to communicated-information. That is, the HIMARS would actually move quickly towards where the target is and when it was within its limited throwing range it would fire a munition. When no enemy agents were present in its sensor range and no information was being received from the forward observers it would then quickly retreat to its initial position.

The problem in using this representation is that enemy agents would react to the HIMARS when it was in their sensor range. Of course ideally the HIMARS would be located stationary at the rear but it was hoped that the high movement range and ability of the HIMARS to advance and retreat so quickly would minimise this unwanted behaviour.

Another feature that cannot be modelled directly in EINSTEIN is a time lag between rounds fired. A weapon such as HIMARS requires a significant time between rounds to reload and acquire a target. It was found that a high Probability-of-Hit value for HIMARS was too lethal, while a lower value of 0.4 provided more realistic behaviour and could be viewed as a form of time delay between rounds.

### 4.1.2 Results

#### 4.1.2.1 Interactive Playback Mode

Interactive Playback mode enables the analyst to examine the behaviour of the entities, which should be correlated with their desired characteristics and tasks. A degree of fine-tuning of the entity parameters is generally required to produce a baseline scenario with all entities functioning in a representative and consistent way. However, one should try to avoid tweaking the parameters unnecessarily in an effort to produce the 'correct behaviour', that is, to produce scripted behaviour. The central point of ABD's is to seek emergent behaviour from the local interaction rules we define—not to constrain that behaviour.

Once the fine-tuning has been performed and a baseline scenario constructed, the Interactive Playback mode allows the analyst to obtain qualitative information about the

force mix dynamic interactions. For our baseline scenario, Red travels tightly grouped from East to West through the AO patrolled by the LAV squad. The LAV, due to their superior sensors and speed, detect the incoming T-80 and communicate these detections back to the waiting ARH and HIMARS. From the ensuing engagements we note that most LAV manage to avoid decisive engagement with the T-80 and generally survive. The Red force is heavily depleted, mainly by the ARH and HIMARS and only a few Red elements manage to reach the objective (represented by the Blue flag).

Thus for the baseline scenario, at least on a qualitative level, it is possible for a smaller, more mobile force with a high degree of situational awareness and effective reach-back munitions to defeat a much larger opposing force. The issue that then arises is the relative contribution to this success of differing force mixes and varying asset characteristics.

#### *4.1.2.2 One-way Sensitivity Analysis*

One-way sensitivity analysis allows the relative effect of individual parameters on the mission to be quantified. As an example of this parameter excursion, we investigated the effect of different force mixes (in terms of the number of ARH and whether or not HIMARS was available) on the success rate of the Red force. The measure of effectiveness (MOE) used was the percentage of Red forces that manage to reach the objective (Blue flag).

Figure 21 shows the variation of this MOE with different numbers of ARH — the upper curve represents the situation with no HIMARS while the lower curve is the case with a single HIMARS unit. With no HIMARS and no ARH the Red force easily achieves its mission, with all entities reaching the objective. With a single HIMARS and no ARH just over half of the Red force now manage to reach the objective. In both cases, as the number of ARH is increased, Red mission success is diminished.

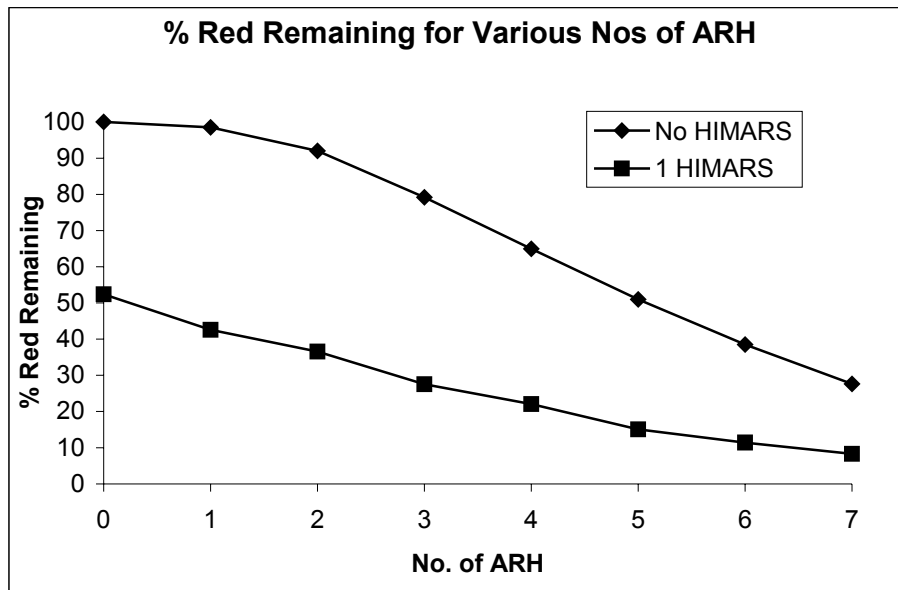


Figure 21 Snapshots of baseline scenario simulation

In both cases, there is some non-linearity present in this diminishment, though it is not strong. In the case of no HIMARS, it appears that at least two ARH are required to significantly affect Red's mission. Also, in the case where there is 1 HIMARS, there appears to be diminishing returns as more and more ARH are added to the force mix. This may suggest that there is an upper limit of ARH that a cost-effective Blue force mix should possess.

Figure 21 can be used to make capability comparisons. For example, the data indicates that to ensure that only 50% of the Red force achieves their objective, this effect could be equally generated with either one HIMARS or six ARH. Similarly, to ensure that only 30% of the Red force achieves their objective, this effect could be equally generated with one HIMARS with either four ARH or eight ARH and no HIMARS. Note that this second result does not scale linearly with the first (which would suggest that one HIMARS with four ARH is equivalent to ten ARH). This type of force-mix trade-off analysis could be useful in supporting acquisition decisions once the relative costs of assets are taken into account.

#### 4.1.2.3 Fitness Landscape Analysis

Essentially, fitness landscape analysis is a two-dimensional sensitivity analysis and the surface plotted shows the variation of the selected MOE with two user-specified parameters, which is a useful mechanism to detect allowable trade-offs (essentially contour lines of the plotted surface) as well as synergies between parameters.

Figure 22 examines the variation of the Red to Blue Survival Ratio (a complement to the usual loss-exchange ratio [LER]) to changes in the size of the Red force (ranging from 15 to

60) and to the level of dispersion of the Red entities (ranging from low to high). The latter was modelled by using the Minimum-Distance-to-Friendly meta-personality. Higher values of the MOE indicate improved Red mission success.

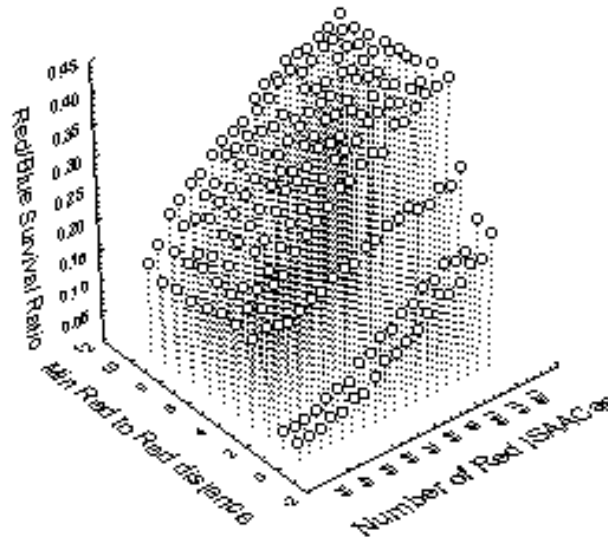


Figure 22 Force size and dispersion level

If we take slices of the surface for different dispersion levels, the shape of the curve is roughly linear with the number of Red forces. Thus, combat weight for Red appears to have a linear effect on success. The surface also clearly shows marked improvement for Red once a dispersion level greater than one is achieved. For dispersion levels greater than three, for a fixed force size, there is no noticeable improvement. Thus, the optimum dispersion level appears to be roughly three, though proper application of ABDs implies that there is a saturation value, rather than a definitive number.

The cause of this result was deduced by running several Interactive Playback sessions, which revealed that the reason is related to the means of employment of the HIMARS. As HIMARS is a limited resource, thresholds were imposed such that delivery of a HIMARS round required a minimum number of enemy targets within a given range and a maximum number of friendly entities. Thus, once Red dispersed to a certain level, it effectively provided Blue with no sufficiently massed target to afford a HIMARS strike by remaining below its engagement threshold.

This result immediately suggests 'what-if' scenarios and ABD's can be used to game these combinations. As mentioned above, this Fitness Landscape analysis can allow trade-offs to be explored. For example, it might be possible for Red to use a smaller but more dispersed force and achieve the same level of mission success.

Figure 23 displays the Fitness Landscape when varying the sensor range and probability of kill (lethality) of Red. Once again, if we examine slices of this landscape for fixed values of the sensor range, we see that the lethality of Red appears to have a linear effect on its mission success.

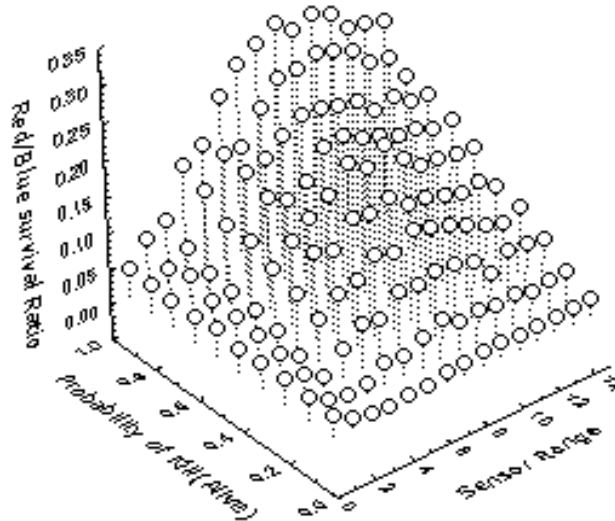


Figure 23 Sensor range and probability of kill

However, the interesting point to note is that the degree of linear effect (essentially the slope of the curve) is not constant but changes quite strongly as the sensor range of Red is increased. Initially this change is positive, whereby the effect of an increase in lethality from 0.4 to 0.6 (for example) is more pronounced with a sensor range of 6 than with a sensor range of 2. This illustrates the potential effect of synergy between platform characteristics.

Note also, however, that this behaviour does not occur for all values of the sensor range, and in fact a reversal of behaviour appears to occur once a sensor range of about 8 is exceeded. On further investigation (by using the Interactive Playback Mode) the cause for this behaviour was traced to the termination criteria of the simulation that produced unrealistic behaviour in those cases.

The goal for the Red force is to reach the Blue objective (the flag) while attempting to minimise its own losses and maximizing losses to the Blue force. The termination criteria used to stop the simulations and collect data on force losses was reaching a fixed time, which needs to be set large enough to allow the mission to be played out. In most cases, the Red force made its way to the objective where it then waited safely until the termination time was reached. However, in the cases where its sensor range was large, it could detect the Blue forces and was drawn back into battle and away from its objective, and suffered increased losses as a result.

Thus the results for these cases should be discarded. However, this analysis is useful in highlighting the need to critically examine the output data and its relevance to the problem under investigation, and the Interactive Playback mode is a useful tool to achieve this.

Again one can also use these landscapes to trade-off parameters, whereby for example the same effectiveness for Red is achieved with a sensor range of 2 and a probability of kill of 1 or a sensor range of 5 and a probability of kill of 0.4. One might suspect that the technological challenges of achieving such a high lethality in the former configuration are such that the latter solution might be more feasible.

#### 4.1.2.4 Dispersion versus Speed

A final trade-off analysis conducted for this scenario was that between the speed of the Red force tanks and the level of dispersion adopted. From the Interactive Playback runs, it is apparent that the casualties suffered by Red occur in the time taken to traverse from its starting position to the objective on the West side of the battlefield. If that time taken could be reduced, then Red would expect to take fewer losses on average.

Thus the situation considered was one of a choice for Red to either conduct its movement along a road or cross-country. The effect of road travel was to increase the speed of the tanks but at the expense of having to travel in a more grouped (or less dispersed) fashion. Cross-country travel was slower but could be performed at different levels of dispersion. Due to the limited number of movement speeds within EINSTEIN, the speed improvement of on-road travel was taken to be a doubling of the cross-country speed.

EINSTEIN was used to produce LER data under three situations – cross country with low dispersion; cross country with medium dispersion; and on road (therefore with no dispersion). Table 9 displays the results generated. Note that a larger LER value corresponds to improved Red performance.

Table 9: LER for different modes of Red movement

Losses	Low Dispersion	Medium Dispersion	On Road
Red Losses	91%	66%	68%
Blue Losses	27%	50%	26%
LER	0.30	0.76	0.38

The results indicate that dispersed travel is preferable if travelling cross-country (which is essentially what the Fitness Landscape in Figure 22 revealed), in that both Red losses are reduced and Blue casualties are increased and the LER is consequently more than doubled. The results also indicate that if travelling on road, then only the Red losses are reduced (by the same margin as dispersed cross country) but the Blue casualties are not

affected. This is because of the decreased time Red has to engage the Blue LAV due to the increased speed on-road, and the decreased ability to hunt the Blue LAV due to being constrained to the road. Consequently there is only a marginal improvement in the LER.

Thus, if only the number of Red losses is important, then both tactics of cross-country dispersed or on-road travel are equally effective. However, if the LER is more important, then the results indicate that the tactic of cross-country dispersed travel would be preferable.

#### 4.1.3 Summary and Conclusions

The case study analysed here produced a number of useful initial insights into the force-mix problem. First, analysis by ABD's allowed quite quickly the contributions of the ARH and HIMARS assets to mission success to be quantified and traded off. The results suggested some regions of non-linearity (decreasing returns) for the ARH effectiveness and highlighted the importance of tactical considerations employed by the Red force against indirect weapons and allowed various tactical options to be evaluated including cross-country or route-movement decisions.

Synergies among platform or weapon characteristics are easily identified using the Fitness Landscape run-mode, and for the force mix problem it was found that sensor range and lethality act quite strongly together. The implication is that investments in weapon and platform upgrades might be best considered jointly rather than in isolation.

ABD's have potential for distilling a problem into the essential elements of the analysis, assuming these components can be modelled to the resolution required of the study. Parameter excursions can easily be conducted (either on PCs running overnight for more reliable statistics, or within say an hour for coarse grained results). This is in stark contrast with traditional war-games whose timescales are measured typically in units of weeks or months.

However, it was found that the EINSTEIN ABD did possess a number of undesirable characteristics. The code is somewhat unstable and some functionality that would have been very useful for the force mix hypothesis studied was either unavailable or did not function properly. For example, the modelling of indirect fires is very limited and some of the purported features associated with terrain did not function as described. Also, there were some variables that would have been quite useful if they were made squad specific, for example, communications range and the selection of targets and the associated lethality against that target.

Finally, it is important to stress that the results of a distillation merely provide some potential directions for further study, which may or may not prove to be useful (depending on the degree of abstraction required to 'fit' an ABD scenario). They do not provide quantitative 'answers'. Their usefulness, if proven to be true, lies in their ability to

quickly provide a focusing of ideas for further higher resolution modelling (for example, in suggesting which factors appear to be important in subsequent war-gaming).

## 4.2 Battlefield Uncertainty

### 4.2.1 Introduction

#### 4.2.1.1 *Background*

Recently, Jan Kuylenstierna, Joacim Rydmark and Tonie Fahraeus, from the Swedish National Defence College conducted experiments using a modified version of chess to explore some of the implications that increases in the level of battlefield uncertainty has on the robustness of various forms of superiority [22].

The question, or hypothesis, they were interested in testing is ‘whether the absolute level of uncertainty about the situation in the battle space is important’, as opposed to the relative level. The experiment was set up with each player using a separate chessboard with a screen between them. A third (impartial) person would make the corresponding move on the opponent’s board. Uncertainty was then modelled using a time lag – that is, players could only see their opponent’s move  $x$  time steps ago (with low values of  $x$  corresponding to minimal uncertainty).

Three edges or superiorities were modelled and analysed. These were, an information edge, which was provided by letting one player see the opponent’s move earlier than they saw their move; a strength edge, which was provided by removing some pieces from one player’s board at the start; and a movement edge, which was provided by allowing one player to make two moves while the opponent made one.

Two general conclusions were made from this study. The conclusion of interest to our study was that tempo was the edge that was more robust to the level of battlefield uncertainty and hence would be the most sought after.

#### 4.2.1.2 *Motivation*

This section contains results from a recent study [23], which outlines an attempt to repeat the chessboard experiments using another analogue of warfare, namely, ISAAC. The motivation for this study was to determine whether the results from the chessboard experiment are independent of the analogue of warfare being used. The author also wanted to utilise the benefits of the ISAAC model to examine the robustness of the chessboard experiment for different scenarios and for different surrogates for uncertainty.

## 4.2.2 Experiment

### 4.2.2.1 *Surrogates for Uncertainty*

As mentioned before, the chessboard experiment utilised a surrogate for battlefield uncertainty whereby each player's knowledge of the location of the opposing forces was delayed by a certain number of moves. Although desirable (in order to closely match their experiment) it was not possible to implement this feature in ISAAC. Thus, instead we examined two alternative surrogates for battlefield uncertainty. This has the advantage of comparing the two experiment's results across these surrogates as well as the model of warfare used.

The first surrogate used was that increased battlefield uncertainty corresponded to decreasing sensor and fire (or weapons) ranges of the individual ISAAC entities. This intuitively feels right if one takes the fog of war literally. It differs from the Swedish experiment in that here, the exact location of some proportion of the opposition's forces is known, whereas in their experiment the approximate (based on where they were exactly some time ago) location of all of the opposition forces was known. For the experiment runs using this surrogate, the data was generated using EINSTEIN.

The second surrogate was proposed after some thought was given to the mechanisms by which the collection of entities would utilise the lower levels of uncertainty (via their increased sensor and weapon ranges). It is generally held that the ability to coordinate action and concentrate force at the correct time is an important factor in generating combat effectiveness. The mechanism to allow this coordination of action between entities within ISAAC is by shared knowledge of the location of the opposition forces.

Under the first surrogate for battlefield uncertainty, this knowledge sharing will only take place when an opposing entity is jointly within the sensor ranges of two or more entities – that is their intersection. This is illustrated in Figure 24 where the proportion or probability of coordinated action by two, three or more entities is relatively small. Taking this diagrammatic argument further, it would appear that improved levels of coordination might result if the knowledge sharing was over the union of the individual sensor ranges. This can be reasonably easily modelled by the use of the communications feature of ISAAC, that is the information of the opposition's locations contained in one entity is communicated to others.

Thus, the second surrogate used was that increased battlefield uncertainty corresponded to decreased weighting to the communicated detections. This indicates the degree to which each side can use information about the opposing forces disposition. For the experiment runs using this surrogate, the data was generated using the facilities at the MHPCC.

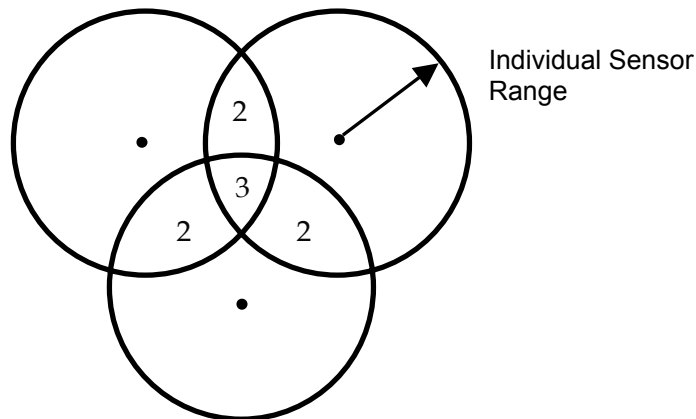


Figure 24 Coordinated actions. Numbers represent the relative probability that two or three entities will respond to the same detection

#### 4.2.2.2 Experiment Design

Two scenarios were played in this experiment – a dispersed scenario and a grouped scenario. This was performed to examine the robustness of the results to scenario variations. It was also conducted since chess would appear to sit perhaps between these two extremes and we wish to contrast across the experiments. Figure 25 shows a screen shot of both scenarios. The dispersed scenario is shown first and it should be noted that the Red and Blue forces are distributed randomly throughout the battlefield. The objective of the dispersed scenario is to manoeuvre in such a way that a numerical advantage can be gained over the opposition and hence be in a position to win the battle. The objective for the combatants in the grouped scenario is to capture the opposing side's flag, which is located diagonally opposite from their respective starting positions.

We attempted to examine the same three edges or superiorities that the chessboard experiment examined – information, tempo and strength. Strength and tempo (or movement) could be modelled in very similar ways. To model strength edge we gave Blue ten percent more entities than Red and to model a tempo edge we allowed Blue to move twice as fast as Red. However, the most feasible way to model the information edge was to increase the sensor range of Blue relative to Red. Unfortunately the sensor range is related to the uncertainty surrogate so the correlation was not as close, and should be considered when analysing the results.

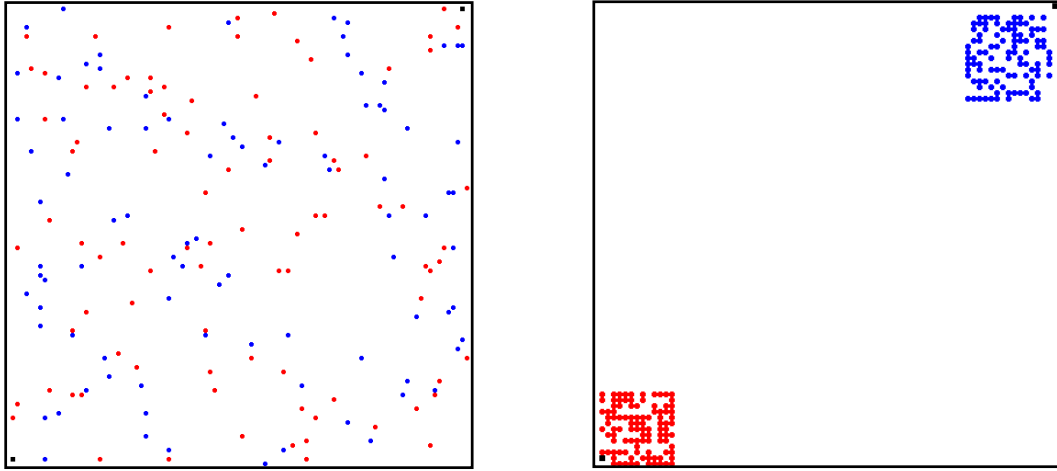


Figure 25 Dispersed Scenario and Grouped Scenario










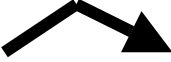


The combination of studying both surrogates for both scenarios and across the three different edges resulted in a total of 12 experiments. Experiments using the sensor/fire range surrogate for uncertainty were run over four different levels of uncertainty ranging from minimal uncertainty (sensor range = 16) to high uncertainty (sensor range = 2). The use of the facilities at MHPCC for the communications weight surrogate allowed us to vary the uncertainty level from 0.1 to 1 in increments of 0.1. For both surrogates 100 runs were conducted at each level of uncertainty and the mean value obtained.

The MOE in chess is ultimately checkmate or resignation although this is more often that not very closely correlated with the relative strengths remaining (a queen sacrifice to force mate in two moves being one exception!) In land warfare, a traditional and generally accepted measure of effectiveness is the LER, which is defined as the number of opposition losses divided by the number of own losses. Thus a LER greater than one indicates better performance. Thus, the MOE used in both experiments are closely related.

#### 4.2.3 Results

The results presented in this section contain no actual figures but concentrates on trends and qualitative analysis. Table 10 allows us to compare edge, surrogate and scenario dependencies and robustness. The direction and shape of the arrows indicates the effect on the LER as the level of uncertainty increases.

Table 10: Effect on LER as uncertainty increases

	Edge >>	Information	Tempo	Strength
	Surrogate			
Dispersed Scenario	Communications Weight			
	Sensor/Fire Range			
Grouped Scenario	Communications Weight			
	Sensor/Fire Range			

Based on the results in Table 10, we can begin to make some conclusions concerning the three edges and their robustness to the level of battlefield uncertainty, the uncertainty surrogate used, and the type of scenario being played. In doing so, we can then make some comparisons with the results and conclusions from the Swedish chessboard experiments.

As mentioned in Section 2.3, the strength and tempo edges were more closely modelled to the chessboard experiment than the information edge. Concerning the strength edge, the results of our ISAAC experiments appear to correlate well with the chessboard experiment, that is as the level of battlefield uncertainty increases the value of an initially superior weight of force diminishes. This conclusion appears to be robust across both the scenario played and the uncertainty surrogate used.

Concerning the tempo edge, the results of our ISAAC experiment appear to partially correlate with the chessboard experiment, in that this edge appears to be robust to battlefield uncertainty but only for the grouped scenario. For the dispersed scenario the results then depend on the uncertainty surrogate used. One could speculate perhaps that this suggests that chess is an example of a grouped scenario.

As mentioned above, the information edge was less well matched between experiments and the results reflect this. However, from our ISAAC experiment we can make the observation that an information edge (based on the communications surrogate) also appears to be robust to both battlefield uncertainty and scenario. This result is contrary to the chessboard experiment, in which the value of an information edge degraded with battlefield uncertainty.

Possibly contributing to this discrepancy is the fact that communications only affects the movement of the agents, and the noted shortcomings of the ISAAC movement algorithm. With the sensor surrogate for uncertainty, the results here also appear to suggest that an optimum level of uncertainty (not equal to the minimum) may exist. This is again somewhat counter-intuitive whereby one would suspect that minimum battlefield uncertainty should provide the best utility.

The results also suggest that the surrogate used for battlefield uncertainty is more important in dispersed scenarios than in grouped scenarios. It is of course impossible to say which surrogate is correct, or even which is more closely representative of battlefield uncertainty. What can be said is that battlefield uncertainty is a more complex concept than that which allows representation by simple surrogates, and that the conclusions based on a single surrogate (as in the chessboard experiments) should be tested and contrasted with others (as we have attempted here).

#### 4.2.4 Conclusions

It should be emphasised that the results presented here are not to be taken as definitive, but rather as providing further information or evidence to support analysis of the effect of battlefield uncertainty (the fog of war) in land warfare. To support this effort further, various avenues of future work should be pursued. A theme that has emerged from this presentation is the need to consider various surrogates for battlefield uncertainty.

Three which are of immediate interest to us are (a) short sighted chess, whereby uncertainty is not modelled by complete information lags, but rather by immediate sensor limitations in much the same way as is modelled in ISAAC; (b) using the command and control structure within ISAAC and a concept of friction which binds subordinates to the local commander; and (c) using higher resolution models of warfare. The drawback of this last suggestion is the overhead in setting up the scenario (which can have timeframes measured in weeks or months) and generating the results (which can have timeframes measured in days or weeks).

We have attempted to repeat the chessboard experiment and therefore have only concentrated on examining the three edges of information, tempo and strength. There are a number of other edges that could be explored, two of which might include lethality (a physical characteristic) and braveness (which could be approximated with behavioural surrogates).

Finally, the results presented here have in the main been based on an examination of the mean value of the distributions. One should utilise more stringent statistical tests of these means, which would allow more rigorous hypothesis testing. Improved statistical analysis might also suggest improved experimental designs, such as factorial designs to reduce the computational effort required for analysis.

## 4.3 Reconnaissance and Surveillance

### 4.3.1 Introduction

Observations made during the Australian Army's HE00 showed that success for the Blue force relied heavily on reconnaissance at the tactical level for collecting and maintaining situation awareness. These observations indicated that the R&S issue was fundamental to the performance of a force conducting manoeuvre operations.

As follow up work [24] a working hypothesis relating battlefield survivability and the effectiveness of R&S was proposed as core question to be explored at the 4<sup>th</sup> Project Albert Workshop, which was held in Cairns, Australia from 6 – 10 Aug 01. This higher-level hypothesis can be broken into a number of more specific lower-level questions. The lower-level questions we attempted to study were:

Q1: 'Does increased SA provide improved survivability?'

Q2: 'What Blue force mix best provides this?'

Q3: 'What capabilities and/or tactics are most critical?'

Q4: 'How does terrain complexity affect Q1 – 3?'

Figure 26 illustrates the baseline scenario elements. The ABD modelling was conducted using the ISAAC model in conjunction with the MHPCC for the data farming. The battlefield consists of a two-dimensional play box of size 150 by 150. The physical and temporal dimensions were a 0.5km grid square and time steps of 1-minute intervals. For the baseline scenario the Blue force consists of 10 high lethality, low protection strike agents and 5 reconnaissance agents, while the Red force consists of 25 high lethality, high protection Red agents.

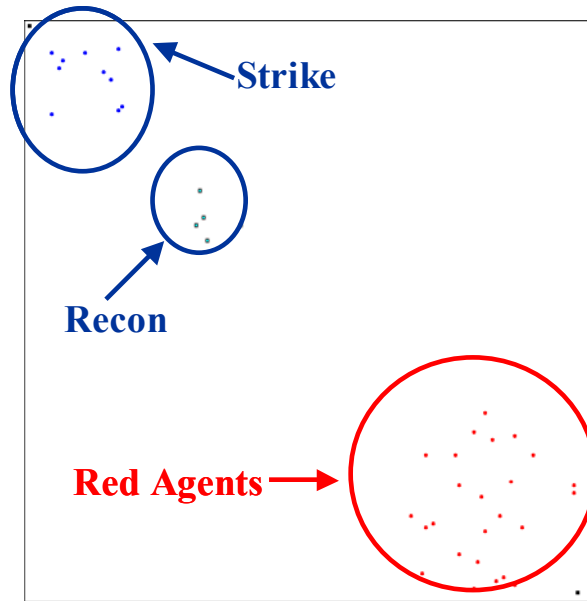


Figure 26 Baseline Scenario Elements

Table 11 shows the excursions from the baseline scenario that was used to investigate the lower-level questions. The two parameters varied were force mix (ORBAT) and terrain complexity. Force mix involved increasing the availability of indirect fire in excursion A, providing more reconnaissance assets in excursion B and finally trading reconnaissance assets for strike assets in Excursion C. Terrain complexity was varied from open to light complexity in alternative 1.

Table 11: Excursions from the Baseline

ORBAT	Terrain Complexity	
	Baseline	Alt1
<b>Baseline</b>	Open	Light
<b>A. Indirect Fire</b>	Open	Light
<b>B. More Recon</b>	Open	Light
<b>C. Traded Recon</b>	Open	Light

The strike agent had a superior sensor range compared to the Red agents but have relatively poorer weapon characteristics. The reconnaissance agents were equipped with 'spotlight' type sensors, faster mobility (thus approximating some form of UAV capability), assumed to be not targeted by Red, and were positioned forward of the strike agents. Their task was to survey the positions of the Red agents and communicate detections back to the strike agents. The strike agents moved towards the Red agents based on the information provided by the reconnaissance agents, and thus relied on good

communications. Tactically, they would engage the Red agents once a numerical advantage was achieved.

Red was effectively static, defending its centre of gravity until Blue units were detected at which time they actively pursue them with an intent to engage. Whilst Red did not require a numerical advantage to attack Blue, they did require at least as many Red units nearby as there are Blue units. Physical attributes were coded in ISAAC using the parameter values given in the left hand side of Table 12 below. The tactical behaviours were modelled in ISAAC as a simple system of attraction-repulsion weightings, and were coded in ISAAC by using the parameter values given in the right hand side of Table 12.

*Table 12: Baseline Model Parameters -- ISAAC Attributes and Personalities*

Parameter	Strike	Recon	Red
<b>Mobility</b>	1	2	1
<b>Sensor</b>	10	4	6
<b>Weapon Range</b>	4	N/A	4
<b>P(Kill)</b>	0.4	N/A	0.5
<b>Defence</b>	2	999	2
<b>Comms Range</b>	80	80	0
<b>Friendly</b>	10	0	0
<b>Enemy</b>	20	50	50
<b>Flag</b>	1	1	0
<b>Combat</b>	5	N/A	0

#### 4.3.2 Workshop Results

The Project Albert workshop format was as follows. The mornings of Day 1 and Day 2 consisted of briefings from each of the syndicate leaders about their respective problem and general presentations about Project Albert research efforts in various nations. The afternoons of Days 1 and 2 were devoted to syndicate discussions, while the evenings allowed opportunities to submit data farming requests to the MHPCC. The morning of Day 3 was used to analyse the results and prepare a back brief that was given by each syndicate leader in the afternoon of Day 3.

During the workshop the baseline and excursions B and C (Table 11) were investigated. The workshop analysis was restricted to the baseline terrain complexity due to limited time. DSTO analysts investigated the remainder of the excursions in Table 11 in a period following the workshop.

The first run was submitted to the MHPCC after 4 hours of scenario planning and discussion at the end of Day 1. Each parameter combination was simulated only 48 times. This value was chosen to ensure that indicative results would be available for analysis on Day 2. For this first excursion (Excursion C, Table 11), we were interested in examining

the force mix of Blue and the strike characteristics. Three Blue parameters were subsequently varied.

1. The number of reconnaissance assets was traded one for one with strike assets.
2. Combat threshold was changed minimum local advantage to attack.
3. Weapon range was changed under match to overmatch.

Figure 27 shows the variation of the percentage (hereafter, expressed as a fraction) of average Red and Blue losses as the number of reconnaissance assets and combat threshold were varied, but holding the Blue weapon range fixed at the baseline value. Both graphs indicate that there seems to be no real trend associated with different values of the combat threshold. This result at first seemed counter-intuitive, but it may well indicate that the combat rule was not often activated because there was minimal dispersion of the Blue agents.

The first graph indicates that there may be a slight optimum, in terms of Red losses, when five reconnaissance assets are present. The second graph is intuitive in telling us that as we trade more vulnerable strike assets for invincible reconnaissance assets we decrease the average number of Blue losses. These results serve to illustrate the ever-present balance between the need for information and a minimum fighting weight.

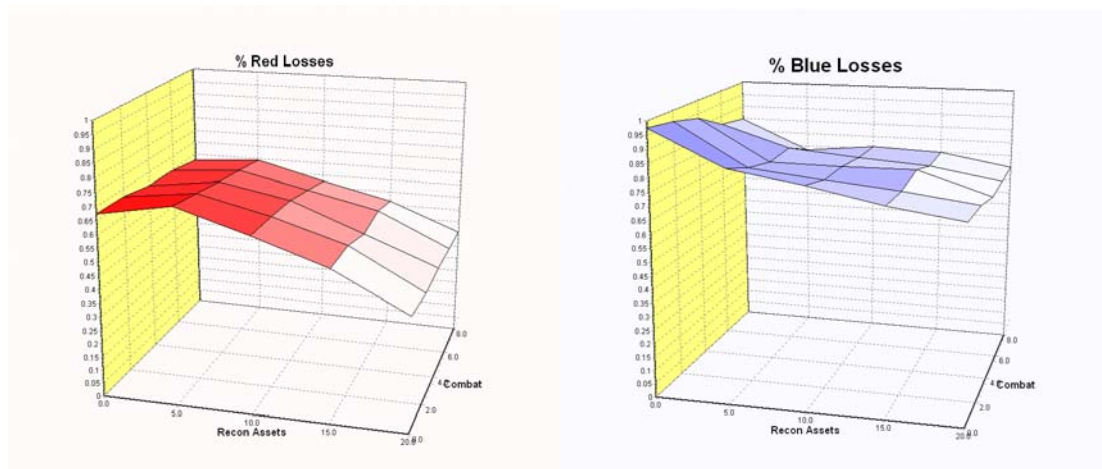


Figure 27 Number of Recon Assets vs. Combat Threshold

Figure 28 shows the variation of the percentage number of average Red and Blue losses as the number of reconnaissance assets and weapon range were varied, but holding the Blue combat threshold fixed at the baseline value. Firstly, we note that as soon as Blue has a weapon range overmatch against Red (i.e.  $> 4$ ) it dramatically increases the average number of Red losses and decreases the average number of Blue losses. Similarly as soon as Blue has a weapon range under match ( $< 4$ ) Red becomes far superior. Hence, relative weapon range appears to be a critical parameter in this scenario. The effect of trading strike assets for reconnaissance assets is minimal except perhaps when there is weapon

range equality where it again appears that a local optimum may exist (as suggested in Figure 24).

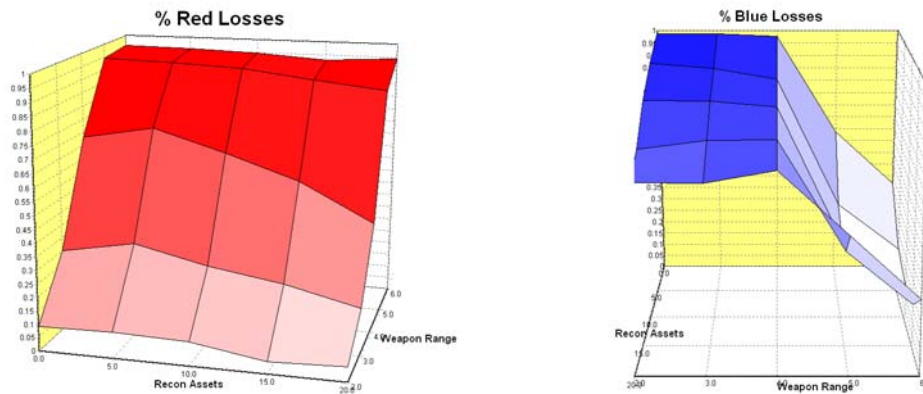


Figure 28 Number of Recon Assets vs. Weapon Range

The second run was submitted to the MHPCC at the end of Day 2 after more scenario planning and discussion and in view of the results from the first run. Each parameter combination was simulated 64 times after information about the total processing time was deduced from the first run. For this second excursion, interest turned to examining force augmentation to Blue and the reconnaissance characteristics (Excursion B, Table 11). Three Blue parameters were subsequently varied:

1. the number of reconnaissance assets<sup>1</sup>
2. communications weight (the 'quality' of the information received)
3. reconnaissance lethality

Initially, the intent was to modify the lethality of just the reconnaissance assets in order to simulate indirect fire. However the data farming tools currently don't allow squad specific changes so unfortunately the lethality of the strike assets was also changed. As a result, this didn't allow us to effectively study the effect of indirect fire during this excursion (Excursion A, Table 11). However, during our follow up data farming we have made an attempt to do this by submitting separate runs – see later.

The reduced set of results for this excursion is given in Figure 29, which shows the variation of the percentage number of average Red, and Blue losses as the number of reconnaissance assets and information quality were varied, but holding the reconnaissance lethality fixed at the baseline value. It should be pointed out that due to model constraints the reconnaissance assets axis is numbered in reverse order so at point 0 there are actually five reconnaissance assets and no reconnaissance assets at point 5. Somewhat surprisingly,

<sup>1</sup> Note: in this scenario the number of reconnaissance assets was increased and the number of strike assets fixed, i.e. there is no trade off.

the results seem to indicate that the effect of increasing the number of reconnaissance assets is minimal, though this rate of return is stronger with increased information quality.

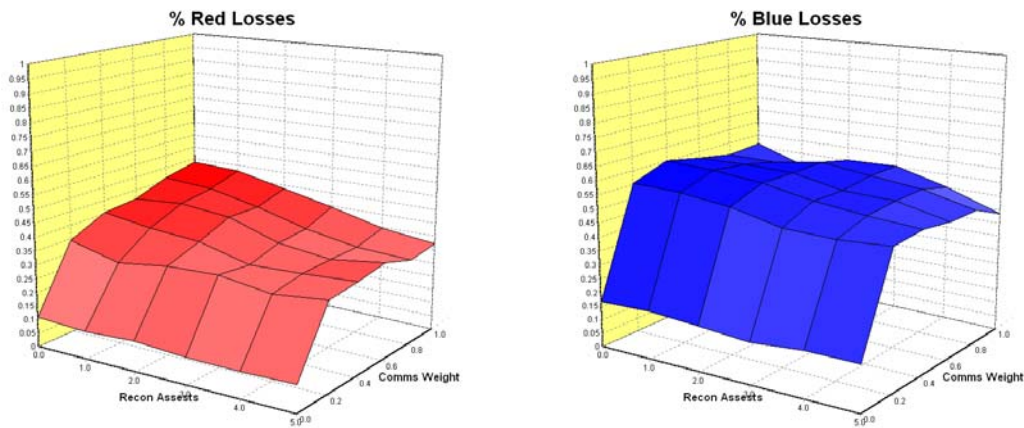


Figure 29 Number of Recon Assets vs. Communications Weight (0-1)

Interestingly, the first graph clearly seems to indicate that there is a significant advantage in increasing the amount of information from none to 'little'. However the second graph indicates that the Blue losses correspondingly increase. The fact that there are very few total losses when Blue has no communications seems to indicate there are very few engagements. As for the reason for a faster rate of increase in Blue losses than Red, the most logical reason is the dispersion of Blue.

With low communications weight there is more dispersion as each entity is acting mainly on their own local information. As communications weight increases the units are gradually drawn to each other, as they all tend to go to areas of high Red concentration making them more grouped and an easier target. Red has no communications and does not suffer from the grouping problem as they act independently. Finally, it would also be interesting to determine whether the 'jump' between 0 and 0.2 on the communications weight axis is gradual or in fact very steep.

#### 4.3.3 Additional Data Farming

In the weeks following the workshop several additional runs were submitted to the MHPCC to further analyse the scenario. The first was alluded to in the previous section and involved homing in on the 'interesting' portion of the graphs from Figure 29. The same two parameters were varied in Figure 30 except that the communications range was only varied between 0 and 0.4 but in smaller increments, and the number of runs for each parameter combination was increased to 96 to produce more reliable statistics.

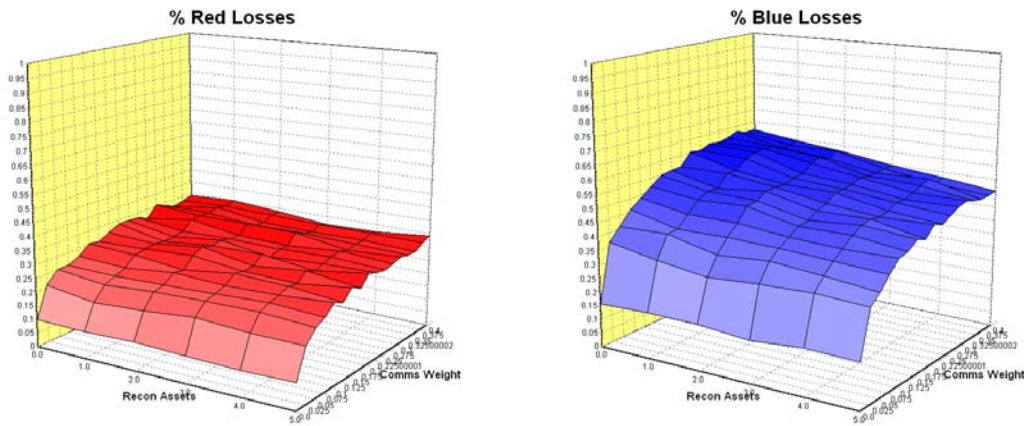


Figure 30 Number of Recon Assets vs. Communications Weight (0-0.4)

The results show that the increase appears to exhibit decreasing returns and thus suggests that there may be little point in improving the quality of information beyond some threshold level (if Red losses is the MOE). This type of analysis may provide insights into answering questions like ‘How much do I need to know about the enemy before I commit my troops’. Alternatively the communications weight parameter could be viewed as a surrogate for the level of trust placed in an entities commander.

In order to examine the robustness of some of the previous results across different terrain types a surrogate for terrain was introduced. In order to simulate a shift from open terrain to light terrain (Alt 1, Table 11) the sensor ranges of all entities were decreased so that the total area sensed was approximately halved in each case. Table 13 details the changes for the respective terrain types.

Table 13: Parameter Modifications for Terrain Type

Open/Light	Strike	Recon	Red
<b>Speed</b>	1 / 1	2 / 2	1 / 1
<b>Sensor</b>	10 / 7	4 / 3	6 / 4
<b>Fire</b>	4 / 4	2 / 2	4 / 4
<b>Lethality</b>	0.25 / 0.25	0 / 0	0.5 / 0/5

Runs were submitted to the MHPCC varying the number of reconnaissance assets from 1 to 10 and simulating the scenario 1024 times for each variation. This illustrates the advantage of models like ISAAC and access to supercomputing facilities in that significantly increased samples sizes can be generated. Figure 31 suggests that in light terrain there may be some form of compromise required when deciding whether to trade strike for reconnaissance (Excursion C, Table 11). As the number of strike agents traded

increases, the number of Red losses decreases (which is undesirable) however the number of Blue losses also decreases (which is desirable).

This raises the question as to which MOE is more important. If both are equally important then the LER could be used to examine whether an optimum exists. The results for the open terrain case show very few losses for both sides. This may suggest that the increased level of awareness on both sides resulted in very few engagements.

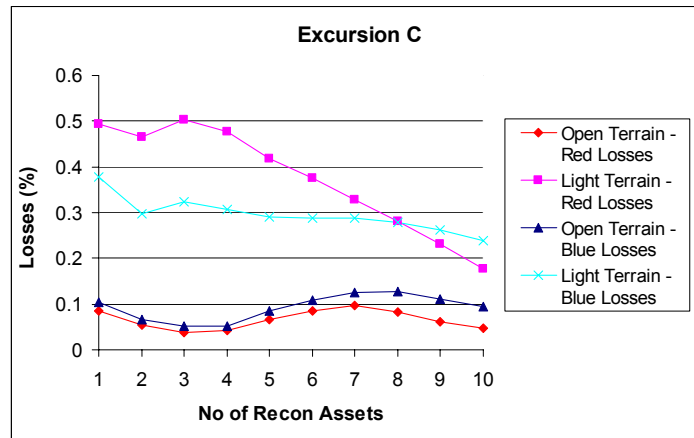


Figure 31 Effect of Trading Strike for Recon Assets in Different Terrain Types

Figure 32 suggests that the number of Red losses appears independent of the number of reconnaissance assets augmented (Excursion B, Table 11) to the Blue force in both open and light terrains. However, the number of Red losses almost doubles in light terrain. This may occur because Red does not possess a stand off capability in light terrain, as modelled here. Red has a sensor range of four and is unable to retreat from Blue before they are fired upon as both sides have a weapon range of four. Blue is unaffected by this stand off problem as their sensor range in light terrain is seven, giving them a sensor overmatch due to their postulated superior technology.

The curves showing average Blue losses indicate a somewhat counter-intuitive result, in that adding reconnaissance agents increases the number of Blue losses (which is the same as the communications result). This result is repeated in both open and light terrain. However, given the earlier result of there being more Red losses in light terrain, it is reasonable to expect that there would then be less Blue losses in light terrain, as the graph indicates.

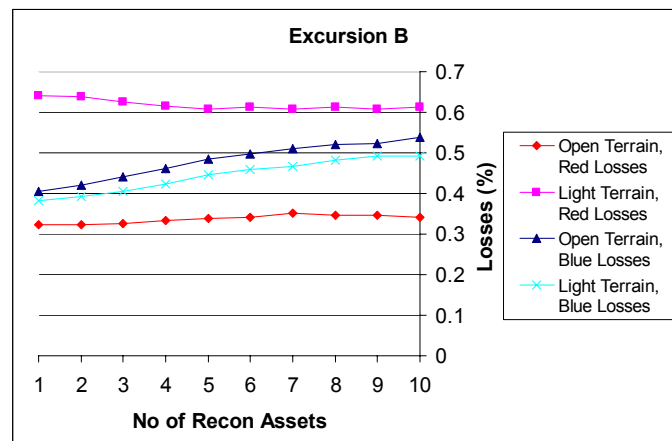


Figure 32 Effect of Adding Strike Assets in Different Terrain Types

As mentioned earlier, in order to simulate indirect fire (Excursion A, Table 11) the reconnaissance agents were given lethality capability and added to a fixed number of strike agents, see Figure 33. When the number of reconnaissance agents was varied from 1 to 10 in light terrain the average number of Red losses increased linearly while the average number of Blue losses decreased linearly, as would be expected with the increased firepower.

However it is interesting to note that in the open terrain scenario the number of Red losses only increases when there is between one and five reconnaissance agents, while after this the number of Red losses remains constant. Meanwhile, Blue losses appear constant regardless of how much indirect fire and reconnaissance assets are present. The combination of all of these results tend to suggest that there is a limit as to how much indirect fire is useful in open terrain, while in light terrain this limit may still be present but the threshold may be much higher.

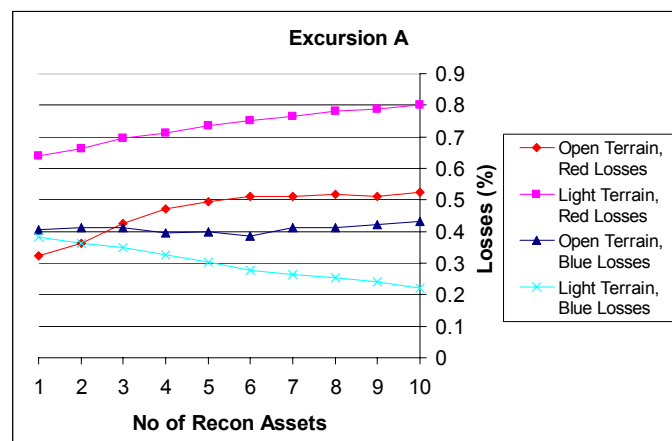


Figure 33 Effect of Adding More Indirect Fire

When reconnaissance and strike assets were traded, the average number of Red losses was very high and virtually constant regardless of how many strike assets were traded for reconnaissance assets. This could be explained by thinking of the trade of reconnaissance assets (with lethality to simulate indirect fires) and strike assets as a trade of firepower for similar firepower. The limitation of the modelling was such that the associated time delay to call in indirect fires (and the resultant decreased in effectiveness) was not modelled. However the number of Blue losses quickly reduced to almost zero as the number of reconnaissance assets increases. This again is intuitive because we are trading relatively unprotected strike assets for highly stealthy, and hence survivable, reconnaissance assets.

#### 4.3.4 Other Modelling

A principle of operational synthesis is the application of a number of analytical techniques to a single problem. CASTFOREM is a closed loop, event driven, stochastic simulation of the combined arms battle, and was able to model the same scenario and excursions as ISAAC in Table 11 with a higher degree of fidelity but with a reduced scope in terms of the farmed parameters. A CASTFOREM study performed before the workshop was thus used to complement the analysis performed by ISAAC. Here, only a summary of the results will be given.

The CASTFOREM results showed that in the baseline scenario Blue has a slight advantage over Red. In Excursion A both the Red and Blue losses increase slightly over the baseline with additional indirect fire assets. However when the number of reconnaissance assets is doubled in Excursion B, Blue inflicts more losses on Red while their Blue losses remain the same as the baseline. Overall they still only have a slight victory. Comparing Excursions A and B suggests that, in this scenario, an investment in reconnaissance and target acquisition has a higher payoff than an investment in indirect fires.

In Excursion C where the reconnaissance assets were traded off for strike assets, there was a similar result to the baseline in terms of losses to both sides and the overall number, resulting in a slight victory for Blue. In this case the model indicated that the force was not sensitive to trading off some strike assets for reconnaissance assets.

However in light vegetation (Alt 1, Table 11) where Red experience a decrease in losses and Blue an increase due to the increase in terrain complexity, Blue suffer a slight defeat in the baseline. In Excursion A and C Blue are able to achieve a slight advantage, hence additional indirect fires or higher levels of reconnaissance (traded for strike in Excursion C) go part way to negating the impact of increased terrain complexity. By far the most interesting result is the impact of additional reconnaissance in Excursion B that leads to a significant victory in light terrain. This result would suggest that additional reconnaissance in combination with the baseline strike assets is a decisive advantage in light terrain. Such a result, while significant in the context of the other results, is the subject of further investigation using higher fidelity terrain models in CASTFOREM.

Table 14 collates the results from the CASTFOREM and ABD models for the various Excursions listed in Table 11. When the results from ISAAC and CASTFOREM are correlated they suggest an investment in reconnaissance or additional strike assets can lead to Blue force being more successful as terrain complexity increases. However analysis from HE00 suggests that this trend would not continue into dense terrain (as opposed to open and light) if the same tactics were maintained. As a result the size of the terrain parameter excursion in the tools was not large enough to synthesis this potentially significant result [24].

Table 14: Comparison of Blue success across Excursions as indicated by the LER

Blue Success / Failure ORBAT	Terrain Complexity			
	Baseline		Alt1	
	CAST	ABD	CAST	ABD
<b>Baseline</b>	☑	☒	☒	☑☑
<b>A. More Indirect Fire</b>	☑	☑	☑	☑☑
<b>B. More Recon</b>	☑	☒	☑☑	☑☑
<b>C. Traded Recon</b>	☑	☒	☑	☑☑

☑ indicates slight Blue victory ( $1.0 < \text{LER} < 1.2$ ), ☑☑ a significant Blue victory ( $\text{LER} > 1.2$ ),  
 ☒ a slight Blue loss ( $0.8 < \text{LER} < 1.0$ ) and ☒☒ a significant Blue loss ( $\text{LER} < 0.8$ )

#### 4.3.5 Conclusions

The workshop clearly showed that the ISAAC model is a useful tool for generating a broad level of discussion amongst defence analysts and military personnel (a tool for thinking with). This enabled both the scenario and concept to be explored and refined in a short period of time. It was found that the visualisation of the two-parameter landscapes was useful to capture both the broad effects (direction and magnitude) of varying single parameters as well as identifying any trade offs and synergies between pairs of parameters.

However, two-parameter landscapes generally don't allow the deduction of causes for these effects. Rather, the interactive nature of the model does allow the suggestion of possibilities (mainly from professional military judgement), which can be useful starting points for other (possibly higher resolution) models in an operational synthesis approach.

The specific insights gained from the ISAAC results have provided trends to compare and contrast with against higher resolution models (e.g. the effectiveness of reach-back), as well as providing a set of precautions (e.g. reconnaissance tactics) to watch out for. Having said this, however, the model and supporting tools still have a number of limitations. In particular, it was found that the modelling of surveillance and intelligence was very

difficult, as was generating variable interaction between force types. Finally, the data farming tools were incapable of ranging over squad specific parameters.

What currently makes ABD's attractive is that they can be applied across a wide range of problem domains, simply and quickly. The approach being pursued by DSTO (Defence systems Analysis Division and Land Operations Division) is to continue to use ABDs as low fidelity tools in an operational synthesis framework to scope a wide problem domain to indicate high payoff analysis areas for higher fidelity tools. Our goal is to improve the strength of the links between ABD's and other tools through the use of system frameworks in which to situate and relate each model and the subsequent results [24].

#### 4.4 Support for Headline Experiment 2001

A team of Project Albert personnel, military subject matter experts, military operations research analysts, and civilian researchers was formed during HE01 in order to support the AEF02 Problem Definition Workshops. In reviewing the initial concepts that pertain to deployment/ redeployment and Entry by the Air and Sea (EAS), it was thought that the present suite of Project Albert models do not readily lend themselves to logistic-based scenarios. Accordingly, the Project Albert team focused their efforts on the EAS perspective.

The aim of the Project Albert team was to employ ABD'S in informing discussion relating to the problem definition workshops [25]. The AEF Study Critical Issue Master List for EAS was reviewed and the areas identified where it was felt that the Project Albert distillations tools would be able to address. The following question was extracted from the master list for analysis:

**Question 1:** What is the relationship/correlation between the number of insertion points and the required weight (force structure / manpower and/or available firepower of the inserted forces at those points)?

It was decided that two of the Project Albert ABD's, MANA and Socrates, might lend themselves to modelling scenarios in addressing this question. However, the limited duration and the intent of HE01 precluded detailed model development, data farming or analysis. Thus the analysis presented below should be viewed as exploratory and the results as indicative.

##### 4.4.1 MANA Results

To explore Question 1 with MANA we considered two broad courses of action (COA) for both Blue and Red, as depicted in Table 15 below. Blue's COA consisted of entering the EAS force by way of a single decisive place (DP) or with multiple DP's. This choice was

made to test the effectiveness or utility of the EAS concept (multiple DP) and to compare it with the more traditional approach (single DP).

Table 15: MANA Scenario Analysis

		RED COA	
		Dispersed	Concentrated
BLUE COA	Single DP	Fail?	Fail?
	Multiple DP	Success?	Fail?

Red's COA were either to disperse his forces across the AO in an attempt to cover all likely points of entry (POE), or to lightly cover a smaller number of POE and maintain a concentrated reserve, who would react to identified Blue entry. These two COA are approximately consistent with those adopted by Red during the wargaming.

The upper portion of Figure 34 below show the initial locations for the single POE scenarios and the two variations in red COA (dispersed and concentrated reserves). The mission for Blue is to sequentially capture the objectives occupied by the three forward Red squads. Red's objective is to repel Blue and defend their area. The reserve is called in to action to assist the top Red squad as soon as any Blue agent is detected.

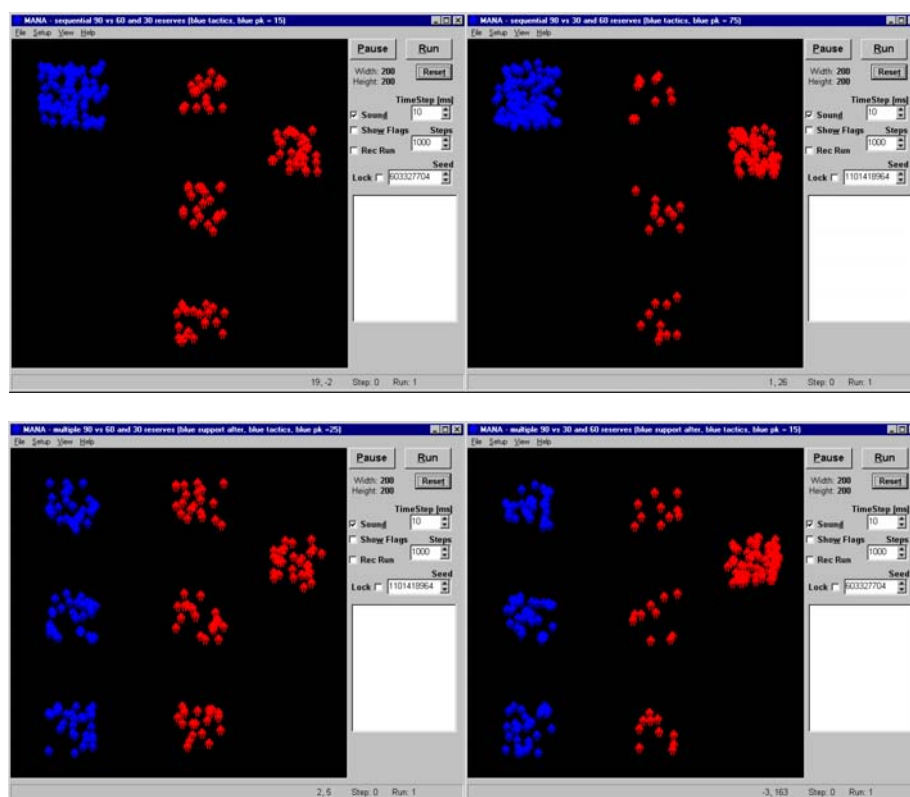


Figure 34 Visual Description of the Four MANA Scenarios

The lower portion of Figure 34 shows the same variations in Red COA but for the multiple POE option. The objective for Blue is to capture the three Red objectives concurrently and to send any remaining forces to the area where the reserves have provided support. In all scenarios the force ratio was kept constant (90 vs. 90) and the physical characteristics (sensor and weapon range, speed, Pk) of Red and Blue were equal except where indicated below.

MANA was run multiple times to generate 50 replications of each of the four vignettes described in Table 15. The mean LER of these 50 replications was used in each case to estimate the performance of the Blue force. It should be noted that the average of 50 replications may not be statistically reliable and that the LER may not be the best MOE for studying the EAS concept. However, both modelling and time constraints have not allowed these limitations to be removed.

The initial findings from the simulations are depicted in Table 16, which provides an indication as to the broad effectiveness of Blue's chosen COA (either single DP or multiple DP) against either Red COA.

*Table 16: Effectiveness of Blue COA against Red COA*

		RED COA	
		Dispersed	Concentrated
BLUE COA	Single DP	2 <sup>nd</sup> Worst	Worst
	Multiple DP	2 <sup>nd</sup> Best	Best

Table 16 indicates that, based on the results of the MANA distillation, the best COA for Blue to adopt is entry by multiple DP (broadly speaking, the EAS concept). Furthermore, this result is independent of which COA Red adopts. Table 16 also indicates that from Red's perspective, his best COA would be to adopt a dispersed posture and not maintain a concentrated reserve. This result is predicated on Red minimising his maximum risk.

In order to help answer the question of the relationship between the number of insertion points and the required weight of forces inserted, we developed three additional scenarios with two, four and five entry points for Blue. We ran these scenarios fifty times each and against both the dispersed and concentrated Red force. The results are summarised in the Figure 35 below.

The general trend indicates that Blue appears to benefit from multiple POE. It is interesting to note that Blue would prefer to challenge a dispersed Red force if one or two forces were inserted. However, when three or more entry points are used the Blue force would prefer to be opposed against a concentrated Red force. In order to further answer the question we need to know if the weight of the inserted force affects these results.

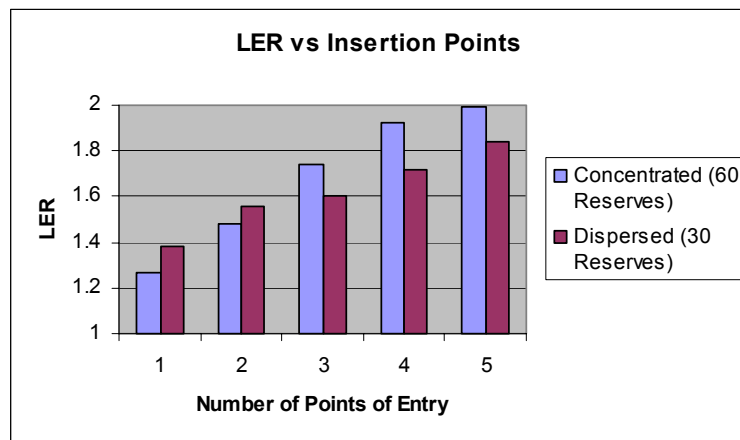


Figure 35 Effectiveness of Increasing Number of POE against Red COA

Figure 36 below shows what happens as we vary one surrogate for force weight (the probability of kill) for Blue. This figure adds robustness to the earlier results in suggesting that more entry points are favourable for Blue across a range of different force weights. In order to conclusively say that this is the case for all of our experiments we would need to make the same changes and run them for the two, four and five entry point options.

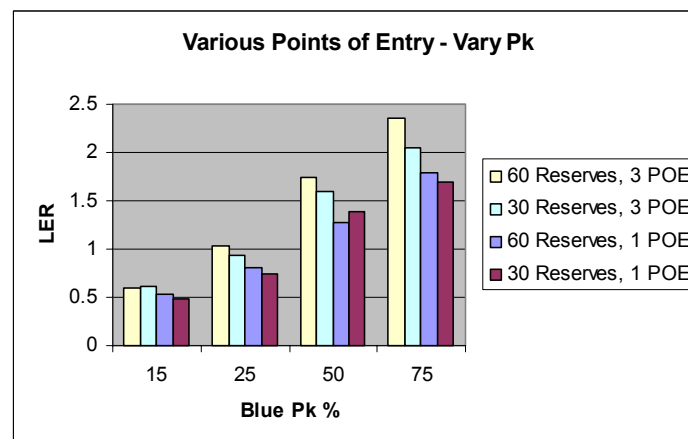


Figure 36 Effect of Firepower Superiority on Best COA

#### 4.4.2 Socrates Results

The efforts of the Socrates modelling group were also focused on Question 1. Two separate scenarios were developed, each with the same Red forces, but with differing configurations of the Blue force. The first scenario employed a single insertion point (see Figure 37a), with the Blue force then progressing onto subsequent objectives. The second

scenario involved the same sized Blue force utilising multiple (three) insertion points (see Figure 37b), attacking the three objectives simultaneously.

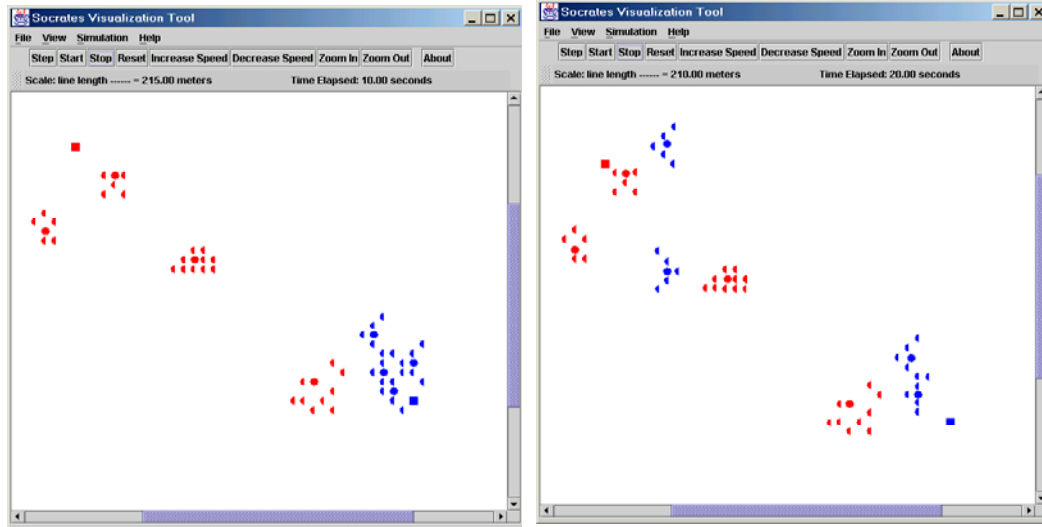


Figure 37 Single and Multiple Insertion Points in Socrates

In both scenarios, the Red force consisted of forces surrounding three objectives and a separate Reserve force. The total force size of the Red force was greater than the Blue, however, the combat effectiveness of each Blue entity was more significant. With the caveat that only a limited number of runs were completed and hence included in the analysis, it was seen that the multiple insertion scenario was preferred as a greater number of Red entities were “neutralised,” with a greater number of Blue forces remaining after the engagements.

The multiple insertion point scenario did not fare as well as the single insertion point scenario, as measured by loss exchange ratio, using the established baseline entity attributes for a 2:3 Blue to Red force ratio. The multiple insertion point scenario was not successful until the Red force commander trust variable was reduced from 1.0 (100% trust) to 0.4 (40% trust).

Due to a limitation in the number of entities that can be inserted into the Socrates model, it was difficult to determine the correlation between the number of insertion points and the required weight of the forces. We compensated by increasing or reducing the capabilities of both Red and Blue forces, to include sensors, weapons, movement and tactics. Though many of the shortfalls created by the limited entity number could be compensated, some could not.

It is considered that further research into relative weapon effectiveness, sensor ranges, stealth, etc., is warranted in order to ascertain the relative weights required to provide the best possible COA for Blue forces. While the initial indication that multiple insertion

points would be preferred to a single point, the relative strength of each scenario must be investigated to determine the sensitivity of these interim results.

#### 4.4.3 Conclusions

ABD's appear to have utility in the formative stages of concept development where broad insights can be gained. For HE01 the use of ABD's could have provided suggestions to both Blue and Red on how to employ the concept or indeed to defeat it. As an example, multiple runs of the MANA ABD demonstrated that even with variations in some of the characteristics of Blue and Red, multiple entry points into the area of operations were more successful than a single point of entry.

This information could have been used by Blue to more properly appreciate the concept prior to subsequent wargaming. To this end ABD's would appear to be more useful if accessed at the earliest opportunity in concept development.

The ABD's available at the time of the analysis do have a number of limitations. For the EAS problem in particular, two are of particular concern. First, the current metric used to measure mission success is the LER. This measure is not ideal when using a behaviour-based model, particularly in an urban environment where non-combatants may be involved. Loss exchange ratios are better suited for linear warfare, not asymmetric warfare. Secondly, the level of intelligence throughout the battlespace is directly related to the accuracy of the model and the degree of confidence of its output. The ABD's available could not, however, measure the level of intelligence (ground, enemy location, intent) required to transition from shape to EAS.

Project Albert's Operational Synthesis methodology, in so far as integrating an additional model type into the traditional toolbox of operations analysis to fill the gap of their collective weaknesses, is certainly sound, and if fully accomplished will add noticeable value to the Army Experimentation Framework in general, and in concept exploration in particular, which is better characterised by uncertainty and a need for scenario-space examination than for high-resolution modelling.

However, from the case studies performed to date it is apparent that an intellectual investment is still required to develop, refine, and analyse both the methodology and the models to transform these from their infancy to maturity. In particular, the current suite of ABD's appear insufficient to address the purported need; a scientific method for data analysis needs to be articulated; and most importantly, the means of integrating the results of ABD modelling with those from the other Operational Synthesis tools needs to be developed and proven to add value.

## 5. Summary

In this section I will briefly comment on which ABD's I have found particularly useful and also give my views on the progress of Project Albert. These comments are my own and are not necessarily shared by others.

The first ABD I started using was EINSTein. Having had no previous experience in any sort of military simulations I thought the package was easy to use and understand. After spending some time modelling some of the scenarios in Section 4 it became clear that there were some limitations and bugs associated with the program. Some of these limitations are addressed by other models (eg multiple trigger states in MANA) while others are not (indirect fire). It is hoped that many of the software bugs may be fixed when the full release is made (currently there only exists a beta version).

I then started to explore the features of ISAAC and was quickly annoyed at the time taken to set up a scenario (compared to EINSTein) and the annoying DOS prompts. I was impressed however by the large amounts of data that could be generated by submitting runs to the MHPCC.

It was about the same time that I started to investigate an early version of MANA. Generally I found it very similar to EINSTein and easy to set up and use. Two features immediately grabbed my attention: the SA map and the trigger states. I personally believe that the SA map has only limited usefulness largely because in a typical scenario it is generally ignored because of the presence of local sensor information (see Section 2.3). The trigger states are extremely useful and I have used them in most scenarios throughout the year. MANA is excellent for qualitative analysis but it is frustrating not to be able to automate parameter excursions.

I then spent some time attempting to learn how to use Archimedes. This was extremely frustrating without any formal documentation and required many phone calls and emails to the developers just to set up a simple scenario. Even then the scenario was only very basic and could not be used to provide any useful information. Soon after the developer's contract expired and we are now awaiting the release of Pythagoras.

The next model to be released was Socrates. Unfortunately other tasks prevented me from spending much time using Socrates except for setting up a simple scenario and investigating it's movement algorithm. If a user is competent in setting up a scenario and understands the movement of agents it appears to be a powerful tool and seems to attempt to address more of the intangible elements of combat.

As mentioned in Section 1.1 Project Albert attempts to investigate operational synthesis and the features of complex adaptive systems. I believe that at times the current suite of ABD's have not been used for this purpose and studies [16, 26] have concentrated more on the physical characteristics of land warfare. It is also important not to draw conclusions

from the results of ABD's alone but to use them to guide further analysis in other models. It is also tempting to take advantage of the computing power available and to produce large amounts of data without really knowing what it is telling us.

On the other hand it is possible to produce large data sets that are totally intuitive or that can already be produced by more traditional models or even using simple equations. A focus on well-defined questions to answer specific needs can help focus data collection to the areas most relevant.

The suite of ABD's has been growing since the formation of Project Albert, new models are being developed and existing models are being extended. Whilst attempts are being made to remove many of the limitations of the current model there is a risk that they may become too detailed and concentrate too much on the physical parameters. If this happens there is a chance that ABD's will become merely an attrition-based model and lose focus on the dynamic interaction between agents.

Traditional models are quite capable of dealing with the physical parameters so I believe that ABD's, as originally intended [2], should focus on the key elements of a complex adaptive system. This way we are achieving part of the aim of operational synthesis by using each tool for what it is particularly useful for.

It is important that the level of abstraction of ABD's is not so great as to prevent realistic comparisons with other models. However, there still needs to be enough detail to allow valid synthesis of information. We have already seen that current ABD's have certain limitations and do not provide enough detail when attempting to replicate scenarios from other models (Section 4.1 and 4.3).

I think that the ideas of Project Albert are excellent and present opportunities for more accurate modelling of real life scenarios. With careful development ABD's may provide valuable insights into the complex adaptive system that the modern battlefield is.

## 6. Acknowledgements

The author thanks Dr Andrew Gill, Dr Richard Egudo, Dr Peter Dortmans, Mr Duncan Tailby, Mr Colin Stanford and Ms Rebecca Cause for their input and support during the year.

## 7. References

1. Brandstein, A. Operational Synthesis: Applying Science to Military Science, Phalanx, Vol. 32, No. 4, 1999.
2. Ilachinski, A. Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Warfare, Center for Naval Analyses, Alexandria, CRM 97-61.10, 1997.
3. Lauren, M.K. and Stephen, R.T. MANA: Map-Aware Non-Uniform Automata. A New Zealand Approach to Scenario Modelling, Journal of Battlefield Technology (in preparation), 2002.
4. Brandstein, A. and Horne, G. Data Farming: A meta-Technique for Research in the 21<sup>st</sup> Century, Maneuver Warfare Science, Marine Corps Combat Development Command Publication, 1998.
5. Sakoda, J.M. 1971. The checkerboard model of social interaction, Journal of Mathematical Sociology, Vol 1, 119 - 132.
6. Berlekamp, E.R., Conway, J.H. and Guy, R.K. Winning Ways for Mathematical Plays, Academic, 1982.
7. Woodcock, A.E.R., Cobb, L. and Dockery J.T. Cellular Automata: A New Method for Battlefield Simulation, Signal, January 1997, 41-50.
8. ISAAC Visualization Toolkit Users Manual, Maui High Performance Computing Centre, 2001.
9. M. Lauren and R. Stephen, Using Fractals to Understand the Role of Entropy in Complexity Science: An Examination of the MANA Combat Model, Fractals (in preparation), 2002.
10. The web site of the Maui High Performance Computer Center, an Air Force Research Laboratory Center managed by the University of Hawai'i, [www.mhpcc.edu](http://www.mhpcc.edu)
11. Ilachinski, A. Enhanced ISAAC Neural Simulation Toolkit (EINSTEIN), An Artificial-Life Laboratory for Exploring Self-Organized Emergence in Land Combat, Beta-Test User's Guide, Center for Naval Analyses, Alexandria, CIM 610.10, 1999.
12. Lauren, M.K. and Stephen, R.T. MANA Map Aware Non-uniform Automata Version 1.0 Users Manual, NZ Defence Technology Agency, 2001.
13. Socrates v1.1 User's Manual, Emergent Information Technologies, Inc., Vienna, VA, 2001.
14. Reynolds, W.N. and Dixon, D.S., Archimedes: A Prototype Distillation, Maneuver Warfare Science, Marine Corps Combat Development Command Publication, 2001.
15. Millikan, J.A., Bactowars: Evolution of Force Mixed Solutions, Presentation given at 4<sup>th</sup> International Project Albert Workshop, August 2001.

16. Gill, A.W., Egudo, R., Dortmans, P.J. and Grieger, D. Supporting the Army Capability Development Process Using Agent Based Distillations – A Case Study, *Journal of Battlefield Technology*, Volume 4 Number 3, 24-29, 2001.
17. Gill, A. and Grieger, D. Validation of Agent Based Distillation Movement Algorithms, DSTO-TN (currently in preparation).
18. Watson, D.F. 1992. *A Guide to the Analysis and Display of Spatial Data*, Pergamon Press, New York.
19. Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. 1989. *Numerical Recipes, The Art of Scientific Computing (Fortran Version)*, Cambridge University Press, Cambridge, 326 – 334.
20. Luce, D. 1959. *Individual Choice Behaviour*, Wiley, New York.
21. Discussions with Roger Stephen, DTA, 2001.
22. J. Kuylenstierna, J. Rydmark and T. Fahraeus, *The Value of Information in War: Some Experimental Findings*, 5th International Command and Control Research and Technology Symposium, Australia War Memorial, Canberra ACT, Australia, 24-26 October 2000.
23. Grieger, D. and Gill, A. W. Exploring the Effect of the Fog of War on the Value of Competitive Edges in Land Warfare, *Computational Techniques and Applications Conference – CTAC01*, (in press), July 2001.
24. Tailby, D., Grieger, D., Gill, A., Stanford, C. and Cause, R. Operational Synthesis for Reconnaissance and Surveillance, *Proceedings of the Land Warfare Conference 2001*, Sydney.
25. Gill, A. and Grieger, D. Agent Based Distillation Support to the HE01, DSTO-GD (currently in preparation).
26. M. Lauren and D. Baigent, Exploring the value of sensors to a Recce Unit using Agent Based Models, *Journal of Battlefield Technology*, Vol. 4, No. 1, pp. 1-6, 2001.

## DISTRIBUTION LIST

### Using Agent Based Distillations to Support Land Warfare Studies

*Dion Grieger*

Land Operations Division  
Defence Science and Technology Organisation

## AUSTRALIA

### DEFENCE ORGANISATION

**Task Sponsor**                      **Chief, LOD**

#### **S&T Program**

Chief Defence Scientist	}	shared copy
FAS Science Policy		
AS Science Corporate Management		
Director General Science Policy Development		
Counsellor Defence Science, London (Doc Data Sheet)		
Counsellor Defence Science, Washington (Doc Data Sheet)		
Scientific Adviser to MRDC Thailand (Doc Data Sheet )		
Scientific Adviser Joint		
Navy Scientific Adviser (Doc Data Sheet and distribution list only)		

Scientific Adviser - Army (Doc Data Sheet and distribution list only)

Air Force Scientific Adviser  
Director Trials

#### **Systems Sciences Laboratory**

Research Leader Operations Analysis and Evaluation (Doc Data Sheet and Distribution Sheet Only)

Head Military Evaluation  
Task Manager: Dr Peng Shi

#### **Information Sciences Laboratory**

Chief of Defence Systems Analysis Division  
Research Leader MSEB  
Head of Group MSEB  
Dr Andrew Gill (2 copies)

#### **DSTO Library and Archives**

Library Edinburgh 2 copies  
Australian Archives

**Capability Systems Staff**

Director General Maritime Development (Doc Data Sheet only)  
Director General Aerospace Development (Doc Data Sheet only)

**Knowledge Staff**

Director General Command, Control, Communications and Computers (DGC4)  
(Doc Data Sheet only)

**Army**

ABCA National Standardisation Officer, Land Warfare Development Sector,  
Puckapunyal (4 copies)  
SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), Enoggera QLD  
(Doc Data Sheet only)

**Intelligence Program**

DGSTA Defence Intelligence Organisation  
Manager, Information Centre, Defence Intelligence Organisation

**Defence Libraries**

Library Manager, DLS-Canberra  
Library Manager, DLS - Sydney West (Doc Data Sheet Only)

**UNIVERSITIES AND COLLEGES**

Australian Defence Force Academy  
Library  
Head of Aerospace and Mechanical Engineering  
Hargrave Library, Monash University (Doc Data Sheet only)  
Librarian, Flinders University

**OTHER ORGANISATIONS**

National Library of Australia  
NASA (Canberra)  
AusInfo  
State Library of South Australia

**OUTSIDE AUSTRALIA****INTERNATIONAL DEFENCE INFORMATION CENTRES**

US Defense Technical Information Center, 2 copies  
UK Defence Research Information Centre, 2 copies  
Canada Defence Scientific Information Service, 1 copy  
NZ Defence Information Centre, 1 copy

**ABSTRACTING AND INFORMATION ORGANISATIONS**

Library, Chemical Abstracts Reference Service  
Engineering Societies Library, US  
Materials Information, Cambridge Scientific Abstracts, US  
Documents Librarian, The Center for Research Libraries, US

**INFORMATION EXCHANGE AGREEMENT PARTNERS**

Acquisitions Unit, Science Reference and Information Service, UK

Library - Exchange Desk, National Institute of Standards and Technology, US

SPARES (5 copies)

**Total number of copies:**